API Gateway

Version 7.6.2

14 July 2020

Upgrade Guide

This documentation describes the following Axway software:

Axway API Gateway 7.6.2

Contents

Preface **8**
Who should read this guide 8
How to use this guide 8
Related documentation 9
Support services 9
Training services 10

Accessibility **11**
Screen reader support 11
Support for high contrast and accessible use of colors 11

Updates and revisions **12**
Changes in version 7.6.2 12
Changes in version 7.6.1 12
Changes in version 7.6.0 12

1 Upgrade and migration overview **13**
Upgrade and migration process 13
Upgrade commands 14
Utility commands 15
Compare upgrade paths 15
Best practices for test upgrades 15
Resolve upgrade errors and warnings 16

2 Compare upgrades from 7.3.x or 7.4.x with upgrades from 7.5.x or 7.6.x **17**

3 Upgrade from API Gateway 7.3.x or 7.4.x **20**
Before you upgrade from 7.3.x or 7.4.x 20
Checklist for the old API Gateway installation 20
Checklist for the new API Gateway 7.6.2 installation 24
Checklist for upgrades without Apache Cassandra 25
Single-node upgrade example (upgrades from 7.3.x or 7.4.x) 26
Sample API Gateway topology 26
Step 1 – Check the old installation 26
Step 2 – Install API Gateway 7.6.2 27
Step 3 – Check the new installation 27
Step 4 – Run export and upgrade commands 27
Step 5 – Run apply command 28
Step 6 – Verify the upgrade 29

Update your Apache Cassandra database connection details 29
Multi-node upgrade example (upgrades from 7.3.x or 7.4.x) 30
Sample API Gateway topology 30
Step 1 – Check the old installation on each node 31
Step 2 – Install API Gateway 7.6.2 on each node 32
Step 3 – Check the new installation on each node 33
Step 4 – Run export and upgrade on each node 33
Step 5 – Run apply on the first Admin Node Manager (NodeA) 35
Step 6 – Run apply on the other nodes 36
Step 7 – Verify the upgrade 37
After you upgrade from 7.3.x or 7.4.x 37
Upgrade API Gateway projects 37
Update Salesforce connector license 38
Upgrade services 38
Migrate the QuickStart tutorial 39
Add nodes to an Apache Cassandra database cluster for HA 40
Upgrade metrics database with versions earlier than 7.4.0 40

4 Upgrade from API Gateway 7.5.x or 7.6.x 41
Before you upgrade from 7.5.x or 7.6.x 42
Checklist for the old API Gateway installation 42
Checklist for the new API Gateway 7.6.2 installation 44
Upgrade Apache Cassandra 45
Best practice 45
Cassandra upgrade steps – Single-node 45
Cassandra upgrade steps – Multi-node single datacenter 49
Cassandra upgrade – Multi-datacenter 54
Single-node upgrade example (upgrades from 7.5.x or 7.6.x) 54
Sample upgrade topology 54
Summary of steps 55
Multi-node upgrade example (upgrades from 7.5.x or 7.6.x) 56
Sample upgrade topology 56
Summary of steps 57
After you upgrade from 7.5.x or 7.6.x 58
Configure a different Apache Cassandra client port 58
Upgrade API Gateway projects 59
Upgrade services 59
Migrate the QuickStart tutorial 59
Zero downtime upgrade 60
Reference configuration 60
ZDU script package 61
Use the ZDU scripts 63

5 Upgrade API Gateway Analytics 70
Summary of steps 70
Rollback strategy 71
Step 1 – Install API Gateway Analytics 7.6.2 71
Rollback strategy 71
Step 2 – Copy third-party JDBC drivers to the new installation 71
Step 3 – Back up the database in the old installation 72
Rollback strategy 72
Step 4 – Run dbsetup to upgrade the database 72
Rollback strategy 73
Available database upgrades 73
Step 5 – Run upgradeconfig to migrate API Gateway Analytics Entity Store customizations 73
upgradeconfig options 74
Use Policy Studio to change API Gateway Analytics configuration 75
Step 6 – Run configureserver to configure your new API Gateway Analytics Entity Store 75
Rollback strategy 76
Step 7 – Migrate custom reports 76
Modify API Services report for API Gateway Analytics 7.3.x and earlier 76
Migrate other files 77
Step 8 – Stop the old version of API Gateway Analytics and start the new version 77
Step 9 – Enable metrics using managedomain 78
managedomain options 79
Rollback strategy 79
Perform a rollback 80
Further information 80

6 Upgrade your metrics database for API Manager 81
Summary of steps 81
Rollback strategy 81
Step 1 – Back up the metrics database in your old installation 82
Rollback strategy 82
Step 2 – Run dbsetup to upgrade the metrics database (with versions earlier than 7.4.0) 82
Rollback strategy 83
Available database upgrades 83
Step 3 – Enable metrics using managedomain 83
managedomain options 84
Rollback strategy 85
Perform a rollback 85
Further information 85

7 Resolve upgrade issues in Policy Studio 86
Resolve issues during upgrade 86
Resolve issues after upgrade 86

8 sysupgrade command reference 88
export command 88
export command rules 88
export command options 89
Sample export commands 89
export command output 90
upgrade command 90
upgrade command rules 91
upgrade command options 91
Sample upgrade commands 92
upgrade command output 93
apply command 93
apply command rules 94
apply command options 94
Sample apply commands 96
apply command output 97
status command 97
clean command 98
clean command rules 99
clean command options 99
Sample clean command 99

9 sysupgrade error reference 100
Export command errors and warnings 100
Check for customizations to jvm.xml and service.xml 100
Failure to export data from pre-7.2 installations 101
Cassandra configuration 101
Inconsistent groups 102
Upgrade command errors and warnings 102
Admin Node Manager host name is invalid 102
API Manager port changed to HTTPS upgrading from 7.3.1 103
Check if API Manager license required 103
Check if McAfee license required 103
Check if RBAC permissions file has changed 104
Check for corrupt JARs in ext/lib 104
Check for old format WSDLs 104
Check for valid API Gateway license 105
SSL certificates for management traffic regenerated 105
Third-party JDBC JARs 106
Metrics database reconfiguration 106
ActiveMQ directory 106
Check for duplicate API Manager data 108
Check for valid FIPS license 108
Check for Apache Cassandra-backed collections 108

Apply command errors and warnings 109
Incorrect Admin Node Manager host 109

10 Frequently asked questions 110
All upgrades 110
Why would you rerun export? 110
What happens if you change the old API Gateway installation after running export? 110
Why would you rerun upgrade? 111
Why would you rerun apply? 111
Why would you run clean? 111
Single-node upgrades 111
What happens if you rerun export when you have already run apply? 112
What happens if you rerun upgrade when you have already run apply? 112
What happens if you rerun apply? 112
What happens if you run clean? 112
Multi-node upgrades 113
Which is the first Admin Node Manager? 113
What happens if you rerun export on the first Admin Node Manager and have already run apply? 114
What happens if you rerun export on a node that is not the first Admin Node Manager and have already run apply? 114
What happens if you rerun upgrade on the first Admin Node Manager and have already run apply? 115
What happens if you rerun upgrade on a node that is not the first Admin Node Manager and have already run apply? 116
What happens if you rerun apply on the first Admin Node Manager? 116
What happens if you rerun apply on a node that is not the first Admin Node Manager? 117
What happens if you run clean on the first Admin Node Manager? 117
What happens if you run clean on a node that is not the first Admin Node Manager? 118
API Gateway Analytics and metrics database upgrades 118
What should you upgrade first - API Gateway or API Gateway Analytics? 118
Do you need to run managedomain to enable metrics? 119

11 Troubleshoot an upgrade 120
Out of memory error when running upgrade 120
Apply was run before export was run on all nodes 120
Group inconsistency errors 121
KPS data missing after upgrade 121
API Gateways missing from topology after upgrade 122
Get help on sysupgrade commands 123

Preface

This guide describes how to upgrade to API Gateway version 7.6.2 from earlier API Gateway versions.

Note

- Upgrading or installing API Gateway or API Manager on Windows is not supported.
- This guide does not describe how to upgrade API Portal. For more information on upgrading API Portal, see the *API Portal Installation and Upgrade Guide*.
- This guide describes how to upgrade a classic deployment only. It does not describe how to upgrade a container deployment. For information on migrating your upgraded classic deployment to a container deployment, see the *API Gateway Container Deployment Guide*.

Who should read this guide

The intended audience for this guide is system engineers who are responsible for installing, configuring, and maintaining API Gateway.

Before upgrading API Gateway you should have an understanding of API Gateway concepts and features. For more information, see the *API Gateway Concepts Guide*.

How to use this guide

This guide should be used in conjunction with the other guides in the API Management documentation set.

Before you begin installing API Gateway, review this guide thoroughly. The following is a brief description of the contents of each section:

- *Upgrade and migration overview* on page 13 – Provides an overview of the upgrade process.
- *Compare upgrades from 7.3.x or 7.4.x with upgrades from 7.5.x or 7.6.x* on page 17 – Compares upgrades from 7.3.x or 7.4.x with upgrades from 7.5.x or 7.6.x.
- *Upgrade from API Gateway 7.3.x or 7.4.x* on page 20 – Describes prerequisites for upgrade, upgrade steps for single-node and multi-node domains, and steps that you must perform after upgrade.
- *Upgrade from API Gateway 7.5.x or 7.6.x* on page 41 – Describes prerequisites for upgrade, upgrade steps for single-node and multi-node domains, and steps that you must perform after upgrade.

- *Upgrade API Gateway Analytics* on page 70 – Describes how to upgrade API Gateway Analytics and the metrics database used for monitoring with API Gateway Analytics.
- *Upgrade your metrics database for API Manager* on page 81 – Describes how to upgrade the metrics database used for monitoring with API Manager or third-party tools.
- *Resolve upgrade issues in Policy Studio* on page 86 – Describes how to use Policy Studio to resolve upgrade issues.
- *sysupgrade command reference* on page 88 – Provides detailed reference information on upgrade commands.
- *sysupgrade error reference* on page 100 – Provides detailed reference information on all upgrade errors and warnings.
- *Frequently asked questions* on page 110 – Provides answers to frequently asked questions about upgrade.
- *Troubleshoot an upgrade* on page 120 – Provides advice on troubleshooting each step in the API Gateway upgrade process.

Related documentation

The AMPLIFY API Management solution enables you to create, publish, promote, and manage Application Programming Interfaces (APIs) in a secure and scalable environment. For more information, see the *AMPLIFY API Management Getting Started Guide*.

The following reference documents are also available on the Axway Documentation portal at https://docs.axway.com:

- *Supported Platforms*

  Lists the different operating systems, databases, browsers, and thick client platforms supported by each Axway product.

- *Interoperability Matrix*

  Provides product version and interoperability information for Axway products.

Support services

The Axway Global Support team provides worldwide 24 x 7 support for customers with active support agreements.

Email support@axway.com or visit Axway Support at https://support.axway.com.

See "Get help with API Gateway" in the *API Gateway Administrator Guide* for the information that you should be prepared to provide when you contact Axway Support.

Training services

Axway offers training across the globe, including on-site instructor-led classes and self-paced online learning. For details, go to: http://www.axway.com/support-services/training

Accessibility

Axway strives to create accessible products and documentation for users.

This documentation provides the following accessibility features:

- *Screen reader support* on page 11
- *Support for high contrast and accessible use of colors* on page 11

Screen reader support

- Alternative text is provided for images whenever necessary.
- The PDF documents are tagged to provide a logical reading order.

Support for high contrast and accessible use of colors

- The documentation can be used in high-contrast mode.
- There is sufficient contrast between the text and the background color.
- The graphics have the right level of contrast and take into account the way color-blind people perceive colors.

Updates and revisions

This guide includes the following documentation changes.

Changes in version 7.6.2

- Updated the topic on upgrading Apache Cassandra to describe how to upgrade from version 2.2.8 to 2.2.12. For more information, see *Upgrade Apache Cassandra* on page 45.
- Restructured the API Gateway Analytics information and added links to the new *API Gateway Analytics User Guide*.

Changes in version 7.6.1

- Updated the guide to clarify that it describes how to upgrade a classic deployment only (and that it does not describe how to upgrade a container deployment).
- Updated the topics on upgrading from 7.5.x to cover upgrading from 7.6.x (for example, 7.6.0) also.

Changes in version 7.6.0

- Updated the guide to recommend that you upgrade Apache Cassandra to version 2.2.8 *before* you upgrade to API Gateway 7.6.2. For more information on upgrading Apache Cassandra, see *Upgrade Apache Cassandra* on page 45.
- Updated the guide to specify that upgrading or installing API Gateway on Windows is not supported.
- Removed references to API Gateway Appliance and API Gateway Analytics, which are no longer provided.
- Updated the existing topic on upgrading API Gateway Analytics to describe how to upgrade your metrics database only, which is still required for monitoring in API Manager. For details, see *Upgrade your metrics database for API Manager* on page 81.

Upgrade and migration overview 1

This topic introduces API Gateway upgrade and migration, and describes how to perform an upgrade of an API Gateway domain from an earlier version to API Gateway version 7.6.2. It also provides recommended best practices for test upgrades.

Note

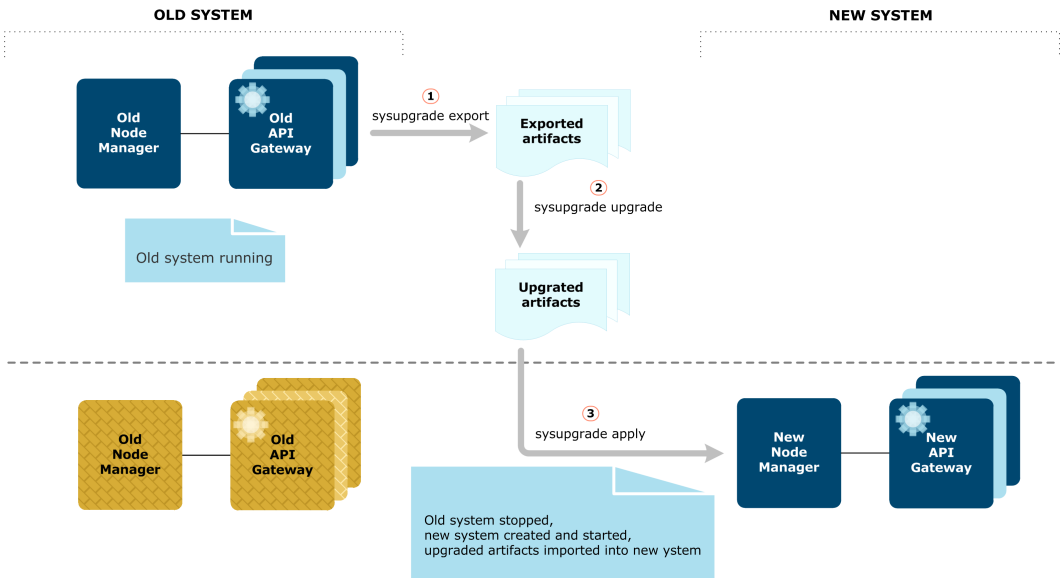
- Upgrading or installing API Gateway or API Manager on Windows is not supported.
- Upgrading or migrating from any previous version of API Gateway requires a new Axway license key.
- This guide describes how to upgrade a classic deployment only. It does not describe how to upgrade a container deployment. For information on migrating your upgraded classic deployment to a container deployment, see the *API Gateway Container Deployment Guide*.
- Throughout this guide the term *old installation* is used to refer to the earlier version of API Gateway (the version being upgraded), while the term *new installation* is used to refer to the 7.6.2 version of API Gateway.

Upgrade and migration process

The upgrade and migration process involves exporting data from your old installation, upgrading the data, and applying the upgraded data to your new installation. Each step must complete successfully before you can proceed to the next step.

Note After you have installed API Gateway 7.6.2, ensure you have installed the latest available service pack before starting the upgrade.

API Gateway provides the `sysupgrade` command to upgrade your API Gateway system and migrate your data. This script provides feedback at each step, which enables you to resolve any issues before proceeding to the next step. This ensures that all possible issues are identified and resolved before you apply the upgraded data to your new installation.



An upgrade always requires the following steps:

1. Install API Gateway 7.6.2 in a new directory.
2. Install the latest available service pack for API Gateway 7.6.2.
3. Export the data from the old installation.
4. Validate and upgrade the data. Resolve any issues identified by the upgrade process before you proceed to the next step.
5. Apply the upgraded data to the 7.6.2 installation.
6. Verify that the upgrade completed successfully.

Upgrade commands

To perform an upgrade, you must always run the following `sysupgrade` commands in the following order:

Step	Command	Description
1	`export`	Export data from the old API Gateway installation on the local node.
2	`upgrade`	Validate data exported from the old installation, and upgrade it to version 7.6.2. You can keep your old installation running during this step, so you can fix any issues reported in the logs without service interruption.
3	`apply`	Create the new API Gateway processes on the local node, and import the upgraded data into the 7.6.2 installation.

Tip You can also use the `status` command at any stage to see which commands have run, and which command will run next (see *Utility commands* on page 15).

For more details on performing an upgrade, see *Upgrade from API Gateway 7.3.x or 7.4.x* on page 20 or *Upgrade from API Gateway 7.5.x or 7.6.x* on page 41.

Utility commands

The `sysupgrade` script also provides utility commands. These optional commands can be useful when performing more complex upgrades:

Command	Description
`status`	Get the status of the `sysupgrade` process on the local node. For example, which commands have run (`export`, `upgrade`, or `apply`), and which command will run next. Running the `status` command is recommended before you run each command, especially in a multi-node upgrade where it is easy to lose track of what upgrade step or machine you are on.
`clean`	Reset the new installation on the current node back to a factory installation. This enables you to restart the `sysupgrade` process.

For details on all commands and options, see *sysupgrade command reference* on page 88.

Compare upgrade paths

For a quick overview of the differences between upgrades from 7.3.x or 7.4.x and upgrades from 7.5.x or 7.6.x, see *Compare upgrades from 7.3.x or 7.4.x with upgrades from 7.5.x or 7.6.x* on page 17.

Best practices for test upgrades

To perform a test upgrade, run the `export` and `upgrade` commands on all nodes in your API Gateway domain. You can do this while the old API Gateway installation is still running, because the `export` and `upgrade` commands do not modify the old installation. This way you can identify and resolve any issues with the upgrade without any service interruption. For example:

1. Run the `export` command on Node1.
2. Run the `upgrade` command on Node1.
3. Analyze the warnings and errors on Node1, and take any necessary actions. You must rerun `export` and `upgrade` after updating your configuration to resolve any warnings or errors. You might need to rerun `export` and `upgrade` multiple times on Node1 to ensure that all issues are resolved.
4. Repeat steps 1, 2, and 3 for Node2, Node3, and so on, until all issues are resolved on all nodes.

As long as `apply` has not been run on any node, you can rerun `export` and `upgrade` as many times as necessary without affecting other nodes. Rerunning `export` and `upgrade` on a node does not mean they must be rerun on other nodes in the system.

When you are happy with the results of the test upgrade on all nodes, you can shut down the old installation and apply the upgraded data to the new 7.6.2 installation. For example:

5. Run the `apply` command on Node1, Node2, and so on, until the upgrade is complete.

For a detailed example of a multi-node upgrade from 7.3.x or 7.4.x, see *Multi-node upgrade example (upgrades from 7.3.x or 7.4.x)* on page 30.

For a detailed example of a multi-node upgrade from 7.5.x or 7.6.x, see *Multi-node upgrade example (upgrades from 7.5.x or 7.6.x)* on page 56.

Resolve upgrade errors and warnings

The `export` and `upgrade` commands identify issues in the configuration that might cause problems during the upgrade. Errors indicate problems that you must resolve before proceeding with an upgrade, while warnings indicate issues that might require action. By identifying and resolving issues at this stage, you can avoid unexpected issues when using the `apply` command to complete the upgrade. For more details on errors and warnings, see *sysupgrade error reference* on page 100.

Compare upgrades from 7.3.x or 7.4.x with upgrades from 7.5.x or 7.6.x 2

This topic compares the steps involved in upgrades from 7.3.x or 7.4.x and upgrades from 7.5.x or 7.6.x. For more details on the steps, see the following examples:

- *Single-node upgrade example (upgrades from 7.3.x or 7.4.x)* on page 26
- *Single-node upgrade example (upgrades from 7.5.x or 7.6.x)* on page 54
- *Multi-node upgrade example (upgrades from 7.3.x or 7.4.x)* on page 30
- *Multi-node upgrade example (upgrades from 7.5.x or 7.6.x)* on page 56

The following table summarizes the differences in the upgrade steps for both upgrade paths.

Steps	Upgrades from 7.3.x or 7.4.x	Upgrades from 7.5.x or 7.6.x
Perform checks on the old installation	If you are using Apache Cassandra: • You must check that embedded Apache Cassandra is configured correctly in your old installation. If you do not wish to use Apache Cassandra in your new installation: • You must check and remove any Cassandra-backed KPS collections in your old installation.	No Apache Cassandra checks are necessary on the old installation, as the external Cassandra configuration is retained when upgrading from 7.5.x or 7.6.x.

Steps	Upgrades from 7.3.x or 7.4.x	Upgrades from 7.5.x or 7.6.x
Install 7.6.2 using the **Custom** installation option	If you are using Apache Cassandra: • To install Apache Cassandra locally on a node alongside API Gateway, select to install the Cassandra component. • To install Apache Cassandra remotely, do not select to install the Cassandra component. • In either case you must configure an Apache Cassandra database cluster in the new installation. If you do not wish to use Apache Cassandra in your new installation: • Do not select to install the Cassandra component.	Do not select to install the Apache Cassandra component, as the external Cassandra configuration is retained when upgrading from 7.5.x or 7.6.x. If you are using Apache Cassandra: • It is recommended that you upgrade Apache Cassandra to version 2.2.12 *before* you upgrade API Gateway.
Perform checks on the new installation	If you are using Apache Cassandra: • You must configure an Apache Cassandra database cluster in the new installation.	You do not need to configure an Apache Cassandra database cluster in the new installation, as the external Cassandra configuration is retained when upgrading from 7.5.x or 7.6.x. If you are using Apache Cassandra: • You must open the port 9042 on your firewall to enable API Gateway to communicate with Apache Cassandra, or you can configure API Gateway to use a different port after the upgrade.

Steps	Upgrades from 7.3.x or 7.4.x	Upgrades from 7.5.x or 7.6.x
Run `export` and `upgrade` commands	If you are using Apache Cassandra: • You must specify the host name or IP address of the *new external Cassandra cluster* to the `upgrade` command using the `--cass_host` option. If you are not using Apache Cassandra: • You must run the `upgrade` command with the `--no_cassandra` option.	If you are using Apache Cassandra: • In a multi-node upgrade, you must specify the host name or IP address of the *existing external Cassandra cluster* to the `upgrade` command using the `--cass_host` option. • In a single-node upgrade, you do not need to specify the host name or IP address of the *existing external Cassandra database cluster* to the `upgrade` command, as the external Cassandra configuration is retained when upgrading from 7.5.x or 7.6.x. If you are not using Apache Cassandra: • In a multi-node upgrade, you must run the `upgrade` command with the `--no_cassandra` option. • In a single-node upgrade, you do not need to run the `upgrade` command with the `--no_cassandra` option.
Run `apply` command	If you are using Apache Cassandra: • You must update `cassandra.yaml` with the host name or IP address of the *new external Cassandra cluster*. • You must start Cassandra before you run `apply`.	If you are using Apache Cassandra: • You do not need to update the `cassandra.yaml` file or start Cassandra, as Cassandra should already be running.

Upgrade from API Gateway 7.3.x or 7.4.x 3

This topic describes how to upgrade from API Gateway 7.3.x or 7.4.x to API Gateway 7.6.2.

Note Upgrading or installing API Gateway or API Manager on Windows is not supported.

This topic contains the following sections:

- *Before you upgrade from 7.3.x or 7.4.x* on page 20
- *Single-node upgrade example (upgrades from 7.3.x or 7.4.x)* on page 26
- *Multi-node upgrade example (upgrades from 7.3.x or 7.4.x)* on page 30
- *After you upgrade from 7.3.x or 7.4.x* on page 37

Before you upgrade from 7.3.x or 7.4.x

This topic describes the steps that you must perform before you upgrade from API Gateway 7.3.x or 7.4.x. It includes checks for your old API Gateway installation and for your new API Gateway 7.6.2 installation. It also includes checks you should perform if you are upgrading without Apache Cassandra. It includes the following sections:

- *Checklist for the old API Gateway installation* on page 20
- *Checklist for the new API Gateway 7.6.2 installation* on page 24
- *Checklist for upgrades without Apache Cassandra* on page 25

Checklist for the old API Gateway installation

You must perform the following in your old API Gateway installation:

- *Back up the old API Gateway installation* on page 21
- *Check that old API Gateway groups are consistent* on page 21
- *Do not update the old API Gateway installation* on page 21
- *Check that embedded Apache Cassandra is configured correctly in the old installation* on page 21
- *Delete or update Apache Cassandra-backed KPS collections in the old installation* on page 25
- *Check that ext/lib customizations in the old installation are compatible* on page 22
- *Update custom filters in API Gateway version 7.3.1 or earlier* on page 22
- *Upgrade API Gateway Analytics version 7.4.0 or later* on page 22

- *Update scripting language filters* on page 22
- *Identify components and configuration requiring manual upgrade steps* on page 23

Back up the old API Gateway installation

Back up the old API Gateway installation on each node. At a minimum:

- Back up the `apigateway` directory.
- Back up any databases used by API Gateway. This includes external databases used for OAuth or KPS, and your metrics database if this has been configured (for example, for monitoring in API Gateway Analytics or API Manager).

For more details on what you should back up, see "API Gateway backup and disaster recovery" in the *API Gateway Administrator Guide*.

Check that old API Gateway groups are consistent

Before you upgrade, ensure that all API Gateway groups in the old installation are consistent, meaning that all API Gateways in a group have the same configuration deployed. Upgrade is not supported for inconsistent groups.

You can use Policy Studio or API Gateway Manager to deploy configuration to API Gateway groups. For more details, see the *API Gateway Administrator Guide*.

Do not update the old API Gateway installation

Do not make any changes to the old API Gateway installation after the upgrade process has begun. For example, if you have run any `sysupgrade` commands, do not perform any of the following on the old installation:

- Do not make any topology changes (for example, add new API Gateway instances)
- Do not deploy any configuration
- Do not update the API Gateway admin user store
- Do not update API Manager configuration (for example, add new APIs, organizations, or applications)

See also *What happens if you change the old API Gateway installation after running export?* on page 110

Check that embedded Apache Cassandra is configured correctly in the old installation

For API Gateway versions with embedded Cassandra (before v7.5.1), you must ensure that embedded Cassandra has been configured correctly in the old API Gateway installation. In particular, there are specific requirements for API Manager high availability (HA). For more details, see "Configure high availability" in the *API Manager 7.4.1 API Management Guide*.

Check that ext/lib customizations in the old installation are compatible

If you have customizations (for example, third-party JAR files) in the `ext/lib` directory of your old API Gateway installation, the `sysupgrade` command copies any third-party JARs to the new installation. However, you must verify that all third-party JARs are compatible with API Gateway 7.6.2.

Before you upgrade, you should confirm that any third-party JARs are not already present in the new installation under directory `apigateway/system/lib`. You should also test any custom JARs to ensure that they work correctly in API Gateway 7.6.2.

For more details, see the *API Gateway Developer Guide*.

Update custom filters in API Gateway version 7.3.1 or earlier

If you are upgrading from API Gateway version 7.2.2 or 7.3.1, and you have developed any custom API Gateway filters in your old installation, you must update your custom filter classes and recompile before upgrading. We recommend that you update your custom filters and recompile them in a 7.6.2 development environment before commencing an upgrade.

The upgrade copies any custom filter classes from `ext/lib` in the old installation to the same location in the new installation. After the upgrade, you must manually remove any incompatible JARs from `ext/lib` in the new installation, and manually copy the recompiled classes to `ext/lib` so that API Gateway can use them. You must also restart any API Gateways that use the custom filters.

For more details on the changes to classes, see the *API Gateway Developer Guide*.

Upgrade API Gateway Analytics version 7.4.0 or later

If you are using API Gateway Analytics version 7.4.0 or later, you can upgrade API Gateway Analytics before you run the `sysupgrade` command. For more information, see *Upgrade API Gateway Analytics* on page 70.

If you are using a metrics database with API Manager and not API Gateway Analytics, see *Upgrade your metrics database for API Manager* on page 81.

Update scripting language filters

If you are upgrading from 7.4.0 or earlier, and you are using a **Scripting Language** filter in your old installation with the **Language** field set to `JavaScript (Rhino engine JRE7 and earlier)`, you must change the **Language** of the filter to `JavaScript` and ensure that the JavaScript syntax in the script conforms with Nashorn engine syntax.

If you do not make these changes, the script continues to work in your new installation, but with a severe drop in performance.

Updating your script to conform with Nashorn engine syntax typically involves replacing the `importPackage` statement, as the following example shows.

Original script:

```
importPackage(Packages.com.vordel.common.base64);

function invoke(msg)
{
    var base64EncodedData = msg.get("data.base64");
    var dataDecoded = Decoder.decodeToString(base64EncodedData);
    msg.put("data.decoded", dataDecoded );
    return true;
}
```

Updated script (complies with Nashorn engine syntax):

```
var base64Import = new JavaImporter(com.vordel.common.base64);
 with(base64Import) {
     function invoke(msg) {
         var base64EncodedData = msg.get("data.base64");
         var dataDecoded = Decoder.decodeToString(base64EncodedData);
         msg.put("data.decoded", dataDecoded);
         return true;
     }
 };
```

For more information about migrating from Rhino to Nashorn, see the Rhino Migration Guide.

Identify components and configuration requiring manual upgrade steps

Not all components and configuration from earlier API Gateway versions can be upgraded automatically with the `sysupgrade` command. However, you can upgrade these items manually.

Check your old installation and identify if you are using any of the following:

- Redaction files – For more information on migrating redaction files, contact Axway Support.
- Customizations to OAuth sample `.md` files – For more information on upgrading these files, contact Axway Support.
- API firewalling – For more information on upgrading API firewalling, contact Axway Support.
- REST APIs developed using the Policy Studio REST API wizard – This wizard was available in earlier API Gateway versions (for example, 7.2.2). For more information on migrating these REST APIs, contact Axway Support.

- Salesforce connector – For more information, see *Update Salesforce connector license* on page 38.
- QuickStart tutorial – For more information on migrating the QuickStart tutorial, see *Migrate the QuickStart tutorial* on page 39.
- API Gateway services – If you are running API Gateway processes as services on Linux, you must upgrade these manually. See *Upgrade services* on page 38.

Checklist for the new API Gateway 7.6.2 installation

Perform the following in your new API Gateway 7.6.2 installation:

- *Install the latest service pack* on page 24
- *Do not start any Node Managers or API Gateways in the new installation* on page 24
- *Configure an Apache Cassandra database cluster in the new installation* on page 24
- *Move third-party JDBC JARs to the new installation* on page 25

Install the latest service pack

Install the latest available service pack for your new installation. Service packs are available from Axway Support at https://support.axway.com.

Do not start any Node Managers or API Gateways in the new installation

Do not create or start any Node Managers, groups, or API Gateways in the new installation. These are started automatically by the `sysupgrade` process.

Configure an Apache Cassandra database cluster in the new installation

Before you run the `sysupgrade apply` step, you must set up an appropriate Apache Cassandra database cluster:

- For upgrade of a single-node or multi-node domain, only one Cassandra server is required in this cluster, and this server receives the upgraded data.
- For upgrade of a multi-node domain, you must not enable authentication on this Cassandra server.
- After the upgrade, you can add more nodes to this cluster to provide high availability (HA), and configure TLS security.

For more information on configuring an Apache Cassandra database cluster, see *Install an Apache Cassandra database* in the *API Gateway Installation Guide*.

Move third-party JDBC JARs to the new installation

If your old API Gateway installation uses external third-party databases for OAuth and KPS, you must copy the JDBC JAR files to the following location in your new 7.6.2 installation:

```
/apigateway/upgrade/lib
```

For example, if your new installation is at `/opt/Axway/7.6.2`, copy the JDBC drivers to `/opt/Axway/7.6.2/apigateway/upgrade/lib`.

This enables the `sysupgrade apply` step to upgrade the databases.

Checklist for upgrades without Apache Cassandra

To upgrade without Apache Cassandra, perform the following in your old API Gateway installation:

- *Delete or update Apache Cassandra-backed KPS collections in the old installation* on page 25

Delete or update Apache Cassandra-backed KPS collections in the old installation

Earlier versions of API Gateway might contain Apache Cassandra-backed KPS collections by default. If you do not wish to install or use Apache Cassandra in your new installation, you must delete these KPS collections, or update them to use an alternative data source, before you upgrade.

Follow these steps:

1. Open the `policystudio.ini` file in your old installation (for example, `/opt/Axway-7.4.1/policystudio/policystudio.ini`) and add the following line:

```
-Dshow.internal.kps.collection=true
```

2. Start Policy Studio and open the project containing the configuration for your old installation.
3. In the Policy Studio tree, navigate to **Environment Configuration > Key Property Stores**.
4. To delete a KPS collection (for example, `API Server`), right-click the KPS collection and select **Delete**.
5. To update a KPS collection to use an alternative data source (for example, a database), click the Browse button next to the **Default Data Source** field and select a new data source.
6. Save and deploy the configuration.

You can now proceed with the upgrade. You must run the `sysupgrade upgrade` step with the `--no_cassandra` option. For more details on this option, see *upgrade command options* on page 91.

Single-node upgrade example (upgrades from 7.3.x or 7.4.x)

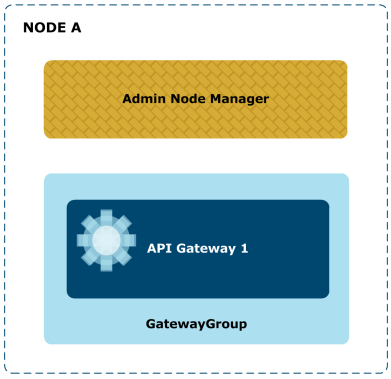
This topic provides a detailed example of how to upgrade an earlier API Gateway version (in this case, 7.4.1) to API Gateway version 7.6.2 in a single-node domain. The `sysupgrade` script includes validation features to guide you through the steps. This topic describes each step in detail.

Tip You can use the steps in this example as a guide when upgrading a single-node domain from API Gateway 7.3.x or 7.4.x to 7.6.2. However, you must remember to modify the steps appropriately for your version and topology.

Sample API Gateway topology

In this example, API Gateway 7.4.1 has a single-node topology that includes one Admin Node Manager and one API Gateway instance. There is a single API Gateway group.

The example topology is shown in the following diagram:



Step 1 - Check the old installation

Perform the checks on your old API Gateway 7.4.1 installation, as detailed in *Checklist for the old API Gateway installation* on page 20.

In addition, if you do not wish to install or use Apache Cassandra in your new installation, perform the checks detailed in *Checklist for upgrades without Apache Cassandra* on page 25.

Step 2 - Install API Gateway 7.6.2

Complete the following steps to install API Gateway 7.6.2:

1. Select the **Custom** option in the installer, and select the following components:
   - Admin Node Manager.
   - API Gateway Server.
   - Policy Studio – Select this only if you want to run Policy Studio on the local machine.
   - API Manager – Select this only if you are upgrading API Manager.
   - Cassandra – Select this only if you want to install an Apache Cassandra database on the local machine alongside API Gateway. Apache Cassandra is required if you are upgrading API Manager, or if you are using Apache Cassandra for custom KPS data, for OAuth client application data, or for API keys in your old API Gateway installation. If you install Cassandra, you will also need to configure a Cassandra cluster. For more details, see *Configure an Apache Cassandra database cluster in the new installation* on page 24.

   Do not select:
   - QuickStart tutorial.

     The QuickStart tutorial creates and starts processes in the new installation. `sysupgrade` requires that no processes are running in the new installation.
2. When prompted for an installation directory, enter a new directory (for example, `/opt/Axway-7.6.2`). A warning message displays if you try to install 7.6.2 in the same directory as the old installation.
3. When prompted to set an administrator user name and password, enter the same Admin Node Manager credentials that you use for the old API Gateway installation.

Tip For more information on installation, including installation prerequisites and how to run the installer in unattended mode, see the *API Gateway Installation Guide*.

Step 3 - Check the new installation

When the installation of API Gateway 7.6.2 is complete, perform the new installation checks detailed in *Checklist for the new API Gateway 7.6.2 installation* on page 24.

Step 4 - Run export and upgrade commands

Perform the following steps:

1. Ensure that the old API Gateway processes are running. This includes, for example, all Admin Node Managers, Node Managers, and API Gateway instances. These processes must be running in the old installation to export the API Gateway configuration data.
2. Change to the `upgrade/bin` directory in the new API Gateway 7.6.2 installation, for example:

```
> cd /opt/Axway-7.6.2/apigateway/upgrade/bin
```

3. Export the data from the old installation, specifying the full path of the old installation. For example, if your old installation is version 7.4.1:

```
> ./sysupgrade export --old_install_dir /opt/Axway-7.4.1/apigateway/
```

   If any issues are identified during export, resolve them and rerun `export` if required.

4. Validate and upgrade the exported data.

   If you are using Apache Cassandra, you must specify the host name or IP address of the new external Cassandra database cluster to the `upgrade` command. For example:

```
> ./sysupgrade upgrade --cass_host 192.0.2.0
```

   For more information, see *Update your Apache Cassandra database connection details* on page 29.

   If you are not using Apache Cassandra, run the `upgrade` command as follows:

```
> ./sysupgrade upgrade --no_cassandra
```

   For more information on the `--no_cassandra` option, see *upgrade command options* on page 91.

If there are any errors or warnings, you are prompted to examine the `sysupgrade` log files. For more details on errors and warnings that can occur during upgrade and recommended actions, see *sysupgrade error reference* on page 100. You can also use Policy Studio to resolve issues with the configuration. For more details, see *Resolve upgrade issues in Policy Studio* on page 86.

You must resolve any errors before proceeding. You can rerun `export` and `upgrade` multiple times until all issues are resolved.

Step 5 - Run apply command

Perform the following steps:

1. Stop the API Gateway processes in the old installation.
2. If you are using Apache Cassandra in the new installation, start Apache Cassandra. Cassandra must be started before running `apply`. For more information on starting Cassandra, see *Install an Apache Cassandra database* in the *API Gateway Installation Guide*.
3. Apply the upgrade to the new installation:

```
> ./sysupgrade apply
```

   This upgrades the external OAuth and KPS databases (if necessary), creates a new system that matches the old topology, and imports the upgraded data.

When all steps have completed successfully, the new API Gateway version 7.6.2 processes should be running.

Note On Linux the Node Manager and API Gateway instances are started by the upgrade process.

Step 6 - Verify the upgrade

To verify that the upgrade has been successful, perform the following steps:

1. Connect to API Gateway Manager (for example, on `https://HOST:8090/`), and view the API Gateway group topology, administrator users, and Key Property Stores. For more details, see the *API Gateway Administrator Guide*.
2. Start Policy Studio, and create a new project based on the running API Gateway. You can view the upgraded configuration (for example, policies, settings, and so on). For more details, see the *API Gateway Policy Developer Guide*.
3. If you were using OAuth client applications in your old installation, start the Client Application Registry web interface, and view the client applications. For more details, see the *API Gateway OAuth User Guide*.

Update your Apache Cassandra database connection details

In API Gateway 7.5.1 and later versions, Cassandra runs externally to the API Gateway process. By default, data for all groups resides in a single Cassandra cluster.

In earlier versions of API Gateway, Cassandra was embedded in the API Gateway process. When you upgrade from an installation with an embedded Cassandra database to an installation with an external Cassandra database, you must run the `sysupgrade upgrade` command with the Cassandra options, to update the Cassandra connection details in the API Gateway configuration. When running `upgrade`, you must specify the new external Cassandra connection details including the Cassandra node IP address and port. You can also specify an optional Cassandra user name and password. For more details, see *upgrade command options* on page 91.

The `sysupgrade apply` command then imports the embedded KPS data from your old installation to the new external Cassandra database cluster. API Gateway group data is placed into a keyspace: `x${`*`DOMAIN_ID`*`}_${`*`GROUP_ID`*`}`, where *`DOMAIN_ID`* is the topology ID of the system being upgraded, and *`GROUP_ID`* is the API Gateway group ID. The replication factor is set to `1`, and read/write consistency levels are set to `ONE/ONE`. This setup enables import into a 1-node or N-node Cassandra cluster, and places data into a known starting point to apply HA and security after upgrade. This supports 1-node (consistent), 2-node, or N-node eventual consistency.

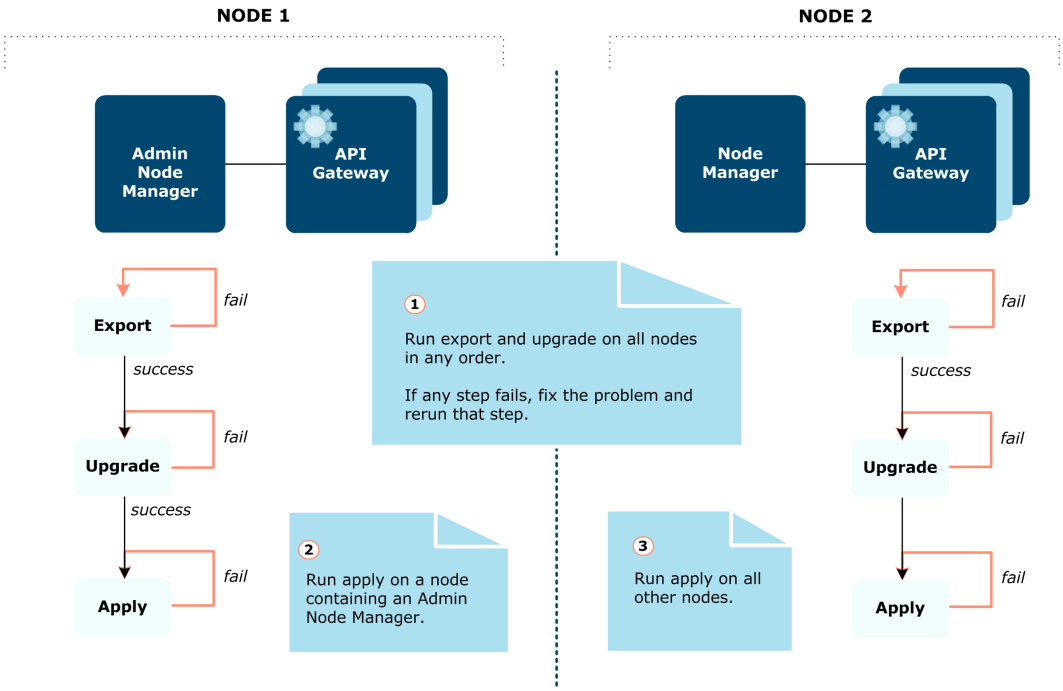
Note If you have not already set up a Cassandra cluster, you must set one up before running the `apply` step. For more details, see *Configure an Apache Cassandra database cluster in the new installation* on page 24. If you rerun `export` after running `apply`, you must first shut down the new Cassandra cluster if it is running on the same hosts as the old API Gateway installation.

Multi-node upgrade example (upgrades from 7.3.x or 7.4.x)

This topic provides a detailed example of how to upgrade an earlier API Gateway version (in this case, 7.4.1) to API Gateway 7.6.2 in a multi-node domain.

Tip You can use the steps in this example as a guide when upgrading a multi-node domain from API Gateway 7.3.x or 7.4.x to 7.6.2. However, you must remember to modify the steps appropriately for your version and topology.

The following diagram shows an example flow for a multi-node upgrade.



The `sysupgrade` script includes validation features to guide you through the steps. We recommend that you use the `status` command to help keep track of the steps on each node.

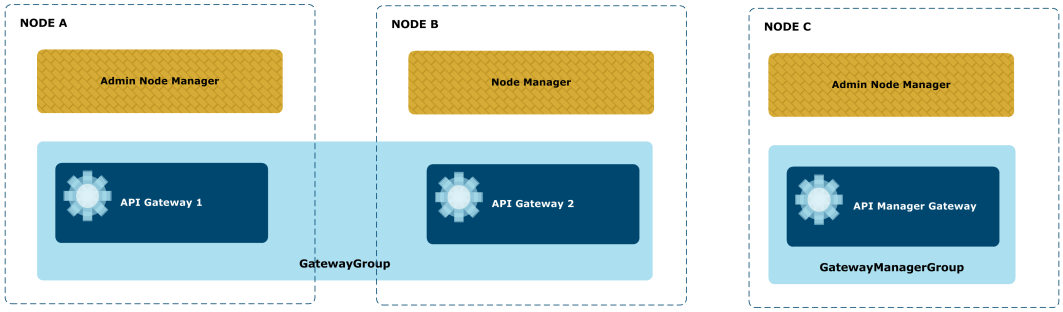
`sysupgrade` does not change the old API Gateway installation. You can always revert back to using the old API Gateway installation.

Sample API Gateway topology

In this example, API Gateway 7.4.1 has a three-node topology that includes two Admin Node Managers (on NodeA and NodeC), and one Node Manager (on NodeB). A single API Gateway instance runs on each node. There are two API Gateway groups as follows:

- `GatewayGroup` has an API Gateway instance running on NodeA and NodeB.
- `GatewayManagerGroup` has a single API Manager-enabled API Gateway instance running on NodeC.

The example topology is shown in the following diagram:



API Gateway NodeA

This node runs an Admin Node Manager and a single API Gateway named `APIGateway1` in a group named `GatewayGroup`.

API Gateway NodeB

This node runs a Node Manager and a single API Gateway named `APIGateway2` in a group named `GatewayGroup`.

API Manager-enabled NodeC

This node runs an Admin Node Manager and a single API Manager-enabled API Gateway named `APIManagerGateway` in a group named `GatewayManagerGroup`.

Step 1 - Check the old installation on each node

Perform the checks on your old API Gateway 7.4.1 installation, as detailed in *Checklist for the old API Gateway installation* on page 20.

Note The sample topology requires Apache Cassandra. If your topology does not use Apache Cassandra, and you do not wish to install or use Apache Cassandra in your new installation, you must also perform the checks detailed in *Checklist for upgrades without Apache Cassandra* on page 25.

Step 2 - Install API Gateway 7.6.2 on each node

On each node in the multi-node topology where your old API Gateway domain is running, you must install API Gateway 7.6.2.

Perform the following steps:

1. Select the **Custom** option in the installer, and select the following components:
    - Admin Node Manager – You must select this on NodeA, NodeB and NodeC for the sample topology.
    - API Gateway Server – You must select this on NodeA, NodeB and NodeC for the sample topology. If you are installing on a node that does not run any API Gateways (running an Admin Node Manager only), you do not need to select this.
    - Policy Studio– Select this on the nodes on which you will run Policy Studio.
    - API Manager– You must select this on NodeC for the sample topology. You can also select it on other nodes if required.
    - Cassandra – Select this only on the nodes on which you want to install an Apache Cassandra database on the local machine alongside API Gateway.

      Do not select Cassandra if you want to install an Apache Cassandra database on a remote machine (on a different host from API Gateway).

      The sample topology requires Apache Cassandra, so you must set up a Cassandra cluster with only one node for upgrade. You must not enable Cassandra authentication on this node. For more details, see *Configure an Apache Cassandra database cluster in the new installation* on page 24.

      For more details on Apache Cassandra, see *Install an Apache Cassandra database* in the *API Gateway Installation Guide*.

   Do not select:

    - QuickStart tutorial.

      The QuickStart tutorial creates and starts processes in the new installation. `sysupgrade` requires that no processes are running in the new installation.
2. When prompted for an installation directory, ensure that you enter a new directory (for example, `/opt/Axway-7.6.2`). A warning message displays if you try to install 7.6.2 in the same directory as the old installation.
3. When prompted to set an administrator user name and password, enter the same Admin Node Manager credentials that you use for the old API Gateway installation.

Tip For more information on installation, including installation prerequisites and how to run the installer in unattended mode, see the *API Gateway Installation Guide*.

Step 3 - Check the new installation on each node

When the installation of API Gateway 7.6.2 is complete, perform the new installation checks detailed in *Checklist for the new API Gateway 7.6.2 installation* on page 24.

Step 4 - Run export and upgrade on each node

You must run the `export` and `upgrade` commands on each node before you can apply the upgrade to the new 7.6.2 system using the `apply` command. You can run and rerun these commands on NodeA, NodeB, and NodeC in any node order.

Identify which Admin Node Manager to specify to export

The sample topology has two Admin Node Managers (running on NodeA and NodeC). Because there are multiple Admin Node Managers, you must specify which Admin Node Manager to use for `export` with the `--anm_host` option, and this Admin Node Manager must also be the first node you upgrade. We recommend that you specify the first Admin Node Manager. For more information, see *Which is the first Admin Node Manager?* on page 113

This example upgrades NodeA first, so you must specify this node using the `--anm_host` option when running `export` on all nodes. The value of `--anm_host` must be an exact match of the host name in the topology (`--anm_host NodeA` in this case) and you must specify the same `--anm_host` on all nodes.

Run export and upgrade

Complete the following steps:

1. Ensure that the old API Gateway processes are running on all nodes. This includes, for example, all Admin Node Managers, Node Managers, and API Gateway instances. These processes must be running in the old installation on all nodes to export the API Gateway configuration data.
2. On each node, run the `export` and `upgrade` commands:

   Change to the `upgrade/bin` directory in the new installation, for example:

```
> cd /opt/Axway-7.6.2/apigateway/upgrade/bin
```

   Export the data from the old installation, for example:

```
> ./sysupgrade export --old_install_dir /opt/Axway-7.4.1/apigateway/ --anm_host
NodeA
```

Note The commands are exactly the same on each node. You must specify `--anm_host Node A` as a parameter to the `export` command on NodeA, NodeB, and NodeC.

If any issues are identified during export, resolve them and rerun `export` if required.

Finally, validate and upgrade the exported data.

The sample topology uses Apache Cassandra to store API Manager data, and you must specify the host name or IP address of the new external Cassandra database cluster to the `upgrade` command. For example:

```
> ./sysupgrade upgrade --cass_host NodeA
```

For more information, see *Update your Apache Cassandra database connection details* on page 29.

Note If you are not using Apache Cassandra, run the `upgrade` command with the `--no_cassandra` option. For more information, see *upgrade command options* on page 91.

If there are any errors or warnings, you are prompted to examine the `sysupgrade` log files. For more details on errors and warnings that can occur during upgrade and recommended actions, see *sysupgrade error reference* on page 100. You can also use Policy Studio to resolve issues with the configuration. For more details, see *Resolve upgrade issues in Policy Studio* on page 86.

You must resolve any errors before proceeding. You can rerun `export` and `upgrade` multiple times until all issues are resolved.

Example export and upgrade errors and solutions

Before running `export` on NodeA, `sysupgrade` checks that the local Admin Node Manager and API Gateway are running and are the expected version. If these checks do not pass, `sysupgrade` generates errors and recommended solutions. For example:

- If the Admin Node Manager is not running:

```
ERROR: Cannot connect to Admin Node Manager on localhost to retrieve product
version.
Please start version [7.4.1] of the Admin Node Manager on localhost.
```

- If the API Gateway is not running:

```
ERROR: Cannot connect to API Gateway [APIGateway1] in Group [Group1] on localhost;
it could be down, or a wrong version could be running.
Please start version [7.4.1] of local API Gateway [APIGateway1] in Group [Group1] on
localhost.
```

- If the Admin Node Manager is running, but it is not the right version:

```
ERROR: Admin Node Manager on localhost is running version [7.3.1], but version
[7.4.1] is required at this point.
Please stop version [7.3.1] of Admin Node Manager on localhost.
Please start version [7.4.1] of Admin Node Manager on localhost.
```

Before running `export` on NodeB, `sysupgrade` checks that the remote Admin Node Manager on NodeA, and the local Node Manager and API Gateway on NodeB, are running and are the expected version. If these checks do not pass, `sysupgrade` generates errors, for example:

- The wrong Admin Node Manager version on NodeA is running:

```
ERROR: Admin Node Manager on host [nodea] is running version [7.3.1], but version
[7.4.1] is required at this point.
Please stop version [7.3.1] of Admin Node Manager on host [nodea].
Please start version [7.4.1] of Admin Node Manager on host [nodea].
```

- Admin Node Manager on NodeA is not running:

```
ERROR: Cannot connect to remote Admin Node Manager on host [nodea].
ERROR: Could not connect to the Admin Node Manager.
```

- Node Manager running on localhost and it is the wrong version, or not running:

```
ERROR: Cannot connect to Node Manager on localhost; it could be down, or a wrong
version could be running.
Please start version [7.4.1] of Node Manager on localhost.
```

Step 5 - Run apply on the first Admin Node Manager (NodeA)

The `export` and `upgrade` steps have now run on all nodes in the topology. The first node on which `apply` runs must be an Admin Node Manager, and because there are multiple Admin Node Managers, you must specify which Admin Node Manager to use for `apply` with the `--anm_host` option. We recommend that you specify the same Admin Node Manager that you specified to the `export` command in *Step 4 – Run export and upgrade on each node* on page 33.

The sample topology has two Admin Node Managers running on NodeA and NodeC. This example runs `apply` on NodeA first, then NodeB and NodeC. In this case, you must specify `--anm_host NodeA` when running `apply` on all nodes. The `--anm_host` value must be an exact match of the host name in the topology and you must specify the same `--anm_host` on all nodes.

To run `apply`, perform the following steps:

1. Shut down all the old API Gateway installation processes on NodeA, NodeB, and NodeC in the topology.
2. If you are using Apache Cassandra, start Apache Cassandra on the host you specified to the `upgrade` command (for example, host `NodeA`). It must be started before running `apply`.

In a multi-node domain, you must make the following changes in the `CASSANDRA_HOME/conf/cassandra.yaml` file before you can start Apache Cassandra:

- `seed provider, parameters, seeds`: Default value is `127.0.0.1`. Change this to the IP address or host name of the Cassandra host.
- `listen_address`: Default value is `localhost`. Change this to IP address or host name of the Cassandra host.
- `rpc_address`: Default value is `localhost`. Change this to IP address or host name of the Cassandra host.

For more information on configuring and starting Cassandra, see *Install an Apache Cassandra database* in the *API Gateway Installation Guide*.

3. Run the `apply` command on NodeA:

```
> cd /opt/Axway-7.6.2/apigateway/upgrade/bin
> ./sysupgrade apply --anm_host NodeA
```

Before running `apply` on NodeA, `sysupgrade` checks that the local Admin Node Manager and API Gateway on NodeA are not running. If you failed to shut down these processes, the following warning appears:

```
Please stop the API Gateway [APIGateway1] in group [GatewayGroup] if it is still
running (management port [8085] is in use)
Please stop the local Node Manager if it is still running (management port [8090] is
in use)
```

The version 7.6.2 Admin Node Manager and API Gateway are now running on NodeA. You can launch the version 7.6.2 API Gateway Manager web console on `https://NodeA:8090`.

Step 6 - Run apply on the other nodes

You can now run `apply` on the other nodes. You must run `apply` on each of the other nodes in turn, and not in parallel.

First, run `apply` on NodeB. For example:

```
> cd /opt/Axway-7.6.2/apigateway/upgrade/bin
> ./sysupgrade apply --anm_host NodeA
```

Before running the `apply` step on NodeB, `sysupgrade` checks the remote Admin Node Manager on NodeA is running and is version 7.6.2. If this check fails, an error similar to the following appears:

```
ERROR: The remote Admin Node Manager on host [NodeA] is version [7.4.1], the
required version is [7.6.2]
On the remote host [NodeA], please ensure that version [7.6.2] of the Admin Node
Manager is running
```

`sysupgrade` also checks that the local Node Manager and API Gateway on NodeB are not running, as shown in *Step 5 – Run apply on the first Admin Node Manager (NodeA)* on page 35.

When `apply` completes on NodeB, run `apply` on NodeC. For example:

```
> cd /opt/Axway-7.6.2/apigateway/upgrade/bin
> ./sysupgrade apply --anm_host NodeA
```

`sysupgrade` is now complete on all nodes. All the API Gateway 7.6.2 processes are running on all nodes in the topology.

Step 7 - Verify the upgrade

Verify the upgrade as detailed in the single-node upgrade example – see *Step 6 – Verify the upgrade* on page 29.

For the sample topology you can also perform the following checks to verify the API Manager upgrade:

1. Connect to the API Manager web console (for example, on `https://HOST:8075/`).
2. Log in as an administrator user and view the organizations, application developers, and applications.
3. Log in as a non-administrator user and view the applications.

For more details on API Manager, see the *API Manager User Guide*.

After you upgrade from 7.3.x or 7.4.x

This topic includes post-upgrade steps that you might need to perform after running `sysupgrade` to upgrade from API Gateway 7.3.x or 7.4.x to 7.6.2. It contains the following topics:

- *Upgrade API Gateway projects* on page 37
- *Update Salesforce connector license* on page 38
- *Upgrade services* on page 38
- *Migrate the QuickStart tutorial* on page 39
- *Add nodes to an Apache Cassandra database cluster for HA* on page 40
- *Upgrade metrics database with versions earlier than 7.4.0* on page 40

Upgrade API Gateway projects

Each API Gateway group has a configuration that is typically deployed as a `.fed` file. When you upgrade from an earlier version of API Gateway, configuration for all API Gateway groups is automatically upgraded during `sysupgrade`. However, you might have configuration files that

were originally created in Policy Studio in a development environment that also need to be upgraded. You can upgrade the configuration in your development environment in one of the following ways:

- In Policy Studio:
    - Choose the **From an API Gateway instance** option to create a new project from the configuration in an already upgraded API Gateway.
    - Choose the **From existing configuration** option to create a new project from an old configuration. The configuration is upgraded to version 7.6.2 automatically.

  For more information on creating projects in Policy Studio, see the *API Gateway Policy Developer Guide*.
- If you upgraded from version 7.4.1 or earlier, you can use the `upgradeconfig` script that is available in your API Gateway installation directory (for example, `/opt/Axway-7.6.2/apigateway/Win32/bin`). For more information on running the `upgradeconfig` script, enter `upgradeconfig --help` at the command prompt.

Caution If your `.fed` file contains API Manager configuration, you cannot upgrade it using `upgradeconfig` or Policy Studio. You must use `sysupgrade`.

Update Salesforce connector license

If you upgraded from API Gateway 7.4.1, the API Gateway trace log might show a deployment error indicating that the Salesforce connector does not have a valid license.

The Salesforce connector was added in API Gateway 7.4.1 and was installed together with API Manager. It can be used to import back-end APIs from Salesforce. In 7.6.2, the Salesforce connector is a licensed feature.

You have two options to resolve this issue:

1. If you intend to use the Salesforce connector in the new API Gateway installation, you must acquire a valid license from Axway Support.
2. If you do not intend to use the Salesforce connector in the new API Gateway installation, you can remove it using Policy Studio. This will eliminate the deployment error.

   For more information on what to remove in Policy Studio, see the Salesforce API connectors section of the *API Manager User Guide*. After removing the connector, you must deploy the updated configuration to all groups in the topology.

Upgrade services

If you were running the API Gateway and Node Manager processes as services in your old installation, you must update the service scripts manually after the upgrade completes. Service scripts are not updated by `sysupgrade`.

Note If you have set up Cassandra to run as a service on the same machine as the API Gateway services, you must ensure that the Cassandra service starts up before the API Gateway services.

Upgrade services on Linux

Complete the following steps after running `sysupgrade apply`:

1. Switch user to `root` to enable you to modify files in `/etc/init.d`. Typically, Axway services file names start with `vshell-`.
2. Edit the Node Manager script and update the `VDISTDIR` variable to point to the `apigateway` folder in the new installation.

   For example, on a machine called XUbuntu02, edit the file `/etc/init.d/vshell-Node-Manager-on-XUbuntu02`.

   - Update the `VDISTDIR` variable (for example, change `VDISTDIR="/opt/Axway-7.2.2/apigateway` to `VDISTDIR="/opt/Axway-7.6.2/apigateway`).
3. Edit each of the relevant API Gateway scripts, and update the `VDISTDIR` and the `VINSTDIR` variables to point to the `apigateway` folder in the new installation.

   For example, on a machine called XUbuntu02 with one API Gateway called `Gateway1` that is a member of a group called `Default Group`, edit the file `/etc/init.d/vshell-Default-Group-Gateway1`.

   - Update the `VDISTDIR` variable (for example, change `VDISTDIR="/opt/Axway-7.2.2/apigateway` to `VDISTDIR="/opt/Axway-7.6.2/apigateway`).
   - Update the `VINSTDIR` variable (for example, change `VINSTDIR="/opt/Axway-7.2.2/apigateway/groups/group-2/instance-1` to `VINSTDIR="/opt/Axway-7.6.2/apigateway/groups/group-2/instance-1`).
4. Save the changes to the files and restart the machine. When the machine restarts the new services are started.

Tip Alternatively, your Linux administrator can remove the old services using the preferred Linux utility and delete the old `init.d` service files, and you can use `managedomain` to recreate the services after running `sysupgrade`.

Migrate the QuickStart tutorial

`sysupgrade` does not migrate the Quickstart tutorial from your old installation. To migrate it, copy the `/apigateway/webapps/quickstart` directory from your old installation (for example, `/opt/Axway/7.4.1/apigateway/webapps/quickstart`) to the same location in the new 7.6.2 installation (for example, `/opt/Axway/7.6.2/apigateway/webapps/quickstart`).

Add nodes to an Apache Cassandra database cluster for HA

Before you upgraded, you configured an Apache Cassandra cluster with one Cassandra server as detailed in *Configure an Apache Cassandra database cluster in the new installation* on page 24. You can now add more nodes to this cluster to provide high availability (HA), and configure TLS security.

For more information, see *Install an Apache Cassandra database* in the *API Gateway Installation Guide*.

Upgrade metrics database with versions earlier than 7.4.0

If you are using an API Gateway version earlier than 7.4.0, you can upgrade your metrics database after you run the `sysupgrade` command. For example, the metrics database is used for monitoring with API Gateway Analytics, API Manager, or third-party tools. For more information, see *Upgrade your metrics database for API Manager* on page 81.

Upgrade from API Gateway 7.5.x or 7.6.x

4

This topic describes how to upgrade from API Gateway 7.5.1 or later to API Gateway 7.6.2.

Note Upgrading or installing API Gateway or API Manager on Windows is not supported.

In API Gateway 7.5.1 and later versions, the Apache Cassandra database is fully separated from the API Gateway (in earlier versions it was embedded with the API Gateway). This simplifies the upgrade process when upgrading from API Gateway 7.5.1 or later, as the data contained in Apache Cassandra does not need to be exported and imported along with the other configuration data.

These are the main differences in an upgrade from API Gateway 7.5.1 and later versions, compared to upgrades from 7.3.x or 7.4.x versions:

- Your existing Apache Cassandra deployment can remain in place for use with API Gateway 7.6.2. There is no need to install a new Apache Cassandra deployment.
- No data changes are necessary in the Apache Cassandra database, which means it can remain running throughout the upgrade, serving any upgraded API Gateways when they come online.

For more information, see *Compare upgrades from 7.3.x or 7.4.x with upgrades from 7.5.x or 7.6.x* on page 17.

Caution

- During an upgrade, including zero downtime upgrade (ZDU), quotas are reset, meaning that your quotas might be too lenient for a period after the upgrade.
- During a ZDU the old quota count and new quota count are in use at the same time.
- Quota could be reduced during ZDU to protect individual instances from being overloaded.

This topic contains the following sections:

- *Before you upgrade from 7.5.x or 7.6.x* on page 42
- *Upgrade Apache Cassandra* on page 45
- *Single-node upgrade example (upgrades from 7.5.x or 7.6.x)* on page 54
- *Multi-node upgrade example (upgrades from 7.5.x or 7.6.x)* on page 56
- *After you upgrade from 7.5.x or 7.6.x* on page 58
- *Zero downtime upgrade* on page 60

Before you upgrade from 7.5.x or 7.6.x

This topic describes the steps that you must perform before you upgrade from API Gateway 7.5.x or 7.6.x. It includes checks for your old API Gateway installation and for your new API Gateway 7.6.2 installation. It includes the following sections:

- *Checklist for the old API Gateway installation* on page 42
- *Checklist for the new API Gateway 7.6.2 installation* on page 44

Note API Gateway 7.6.2 supports Apache Cassandra version 2.2.12. If you are upgrading from API Gateway 7.5.x (which supported Apache Cassandra 2.2.5 and 2.2.8) it is recommended that you upgrade Apache Cassandra to version 2.2.12 *before* you upgrade to API Gateway 7.6.2. For more information on upgrading Apache Cassandra, see *Upgrade Apache Cassandra* on page 45.

Checklist for the old API Gateway installation

You must perform the following in your old API Gateway installation:

- *Back up the old API Gateway installation* on page 42
- *Check that old API Gateway groups are consistent* on page 42
- *Do not update the old API Gateway installation* on page 43
- *Check that ext/lib customizations in the old installation are compatible* on page 43
- *Upgrade API Gateway Analytics version 7.4.0 or later* on page 43
- *Identify components and configuration requiring manual upgrade steps* on page 43

Back up the old API Gateway installation

Back up the old API Gateway installation on each node. At a minimum:

- Back up the `apigateway` directory.
- Back up any databases used by API Gateway. This includes external databases used for OAuth or KPS, and your metrics database if this has been configured (for example, for monitoring in API Gateway Analytics or API Manager).

For more details on what you should back up, see "API Gateway backup and disaster recovery" in the *API Gateway Administrator Guide*.

Check that old API Gateway groups are consistent

Before you upgrade, ensure that all API Gateway groups in the old installation are consistent, meaning that all API Gateways in a group have the same configuration deployed. Upgrade is not supported for inconsistent groups.

You can use Policy Studio or API Gateway Manager to deploy configuration to API Gateway groups. For more details, see the *API Gateway Administrator Guide*.

Do not update the old API Gateway installation

Do not make any changes to the old API Gateway installation after the upgrade process has begun. For example, if you have run any `sysupgrade` commands, do not perform any of the following on the old installation:

- Do not make any topology changes (for example, add new API Gateway instances)
- Do not deploy any configuration
- Do not update the API Gateway admin user store
- Do not update API Manager configuration (for example, add new APIs, organizations, or applications)

See also *What happens if you change the old API Gateway installation after running export?* on page 110

Check that ext/lib customizations in the old installation are compatible

If you have customizations (for example, third-party JAR files) in the `ext/lib` directory of your old API Gateway installation, the `sysupgrade` command copies any third-party JARs to the new installation. However, you must verify that all third-party JARs are compatible with API Gateway 7.6.2.

Before you upgrade, you should confirm that any third-party JARs are not already present in the new installation under directory `apigateway/system/lib`. You should also test any custom JARs to ensure that they work correctly in API Gateway 7.6.2.

For more details, see the *API Gateway Developer Guide*.

Upgrade API Gateway Analytics version 7.4.0 or later

If you are using API Gateway Analytics version 7.4.0 or later, you can upgrade API Gateway Analytics before you run the `sysupgrade` command. For more information, see *Upgrade API Gateway Analytics* on page 70.

If you are using a metrics database with API Manager and not API Gateway Analytics, see *Upgrade your metrics database for API Manager* on page 81.

Identify components and configuration requiring manual upgrade steps

Not all components and configuration from earlier API Gateway versions can be upgraded automatically with the `sysupgrade` command. However, you can upgrade these items manually.

Check your old installation and identify if you are using any of the following:

- Redaction files – For more information on migrating redaction files, contact Axway Support.
- Customizations to OAuth sample `.md` files – For more information on upgrading these files, contact Axway Support.
- API firewalling – For more information on upgrading API firewalling, contact Axway Support.
- QuickStart tutorial – For more information on migrating the QuickStart tutorial, see *Migrate the QuickStart tutorial* on page 59.
- API Gateway services – If you are running API Gateway processes as services on Linux, you must upgrade these manually. See *Upgrade services* on page 59.

Checklist for the new API Gateway 7.6.2 installation

Perform the following in your new API Gateway 7.6.2 installation:

- *Install the latest service pack* on page 44
- *Do not start any Node Managers or API Gateways in the new installation* on page 44
- *Open the new Apache Cassandra client port in the firewall* on page 44
- *Move third-party JDBC JARs to the new installation* on page 45

Install the latest service pack

Install the latest available service pack for your new installation. Service packs are available from Axway Support at https://support.axway.com.

Do not start any Node Managers or API Gateways in the new installation

Do not create or start any Node Managers, groups, or API Gateways in the new installation. These are started automatically by the `sysupgrade` process.

Open the new Apache Cassandra client port in the firewall

API Gateway version 7.6.2 includes the Datastax Cassandra client, which uses a default port of 9042 to communicate with Cassandra over the native protocol. Earlier API Gateway versions included the Hector Cassandra client, which used a default port of 9160 to communicate with Cassandra over the Apache Thrift protocol.

When upgrading from 7.5.1 or later to API Gateway 7.6.2 all Cassandra hosts are updated to use port 9042 for client communication. You must open the port 9042 on your firewall to enable API Gateway to communicate with Apache Cassandra.

Alternatively, to continue to use the same port as you used in your old installation, you can perform some manual steps after the upgrade completes. For more information, see *Configure a different Apache Cassandra client port* on page 58.

Move third-party JDBC JARs to the new installation

If your old API Gateway installation uses external third-party databases for OAuth and KPS, you must copy the JDBC JAR files to the following location in your new 7.6.2 installation:

```
/apigateway/upgrade/lib
```

For example, if your new installation is at `/opt/Axway/7.6.2`, copy the JDBC drivers to `/opt/Axway/7.6.2/apigateway/upgrade/lib`.

This enables the `sysupgrade apply` step to upgrade the databases.

Upgrade Apache Cassandra

This topic explains how to upgrade Apache Cassandra from version 2.2.8 to version 2.2.12. It is recommended that you upgrade your Cassandra version *before* you upgrade your API Gateway installation to version 7.6.2.

Best practice

We recommend the following when upgrading your Apache Cassandra version:

- Upgrade your API Gateway installation after upgrading your Apache Cassandra version.
- In multi-datacenter clusters, upgrade every node in one datacenter before upgrading another datacenter. Upgrade and restart the nodes one at a time. Other nodes in the cluster continue to operate at the earlier version until all nodes are upgraded.
- When upgrading a cluster on a single-datacenter or multi-datacenter setup, you must avoid any schema changes until the entire cluster has been upgraded to the same version.
- Running `nodetool repair` on a Cassandra node will affect performance on a system running live traffic. It is recommended that you perform the Cassandra upgrade in the evening or during a maintenance window when the load is minimal.

Cassandra upgrade steps – Single-node

The following steps give an example of how to upgrade Cassandra in a single-node setup.

Step 1 - Install Cassandra 2.2.12

Follow these steps to install Apache Cassandra 2.2.12 using the API Gateway installer in default GUI mode.

1. Select the **Custom** option in the installer.
2. When prompted to select components to install, select only the **Cassandra** component.
3. When prompted for the Cassandra configuration, enter the following settings:

Installation Directory	JRE Location
/opt/db/cassandra-2212	/opt/jre

4. Do not start Cassandra 2.2.12 when the installation completes. (Your earlier version of Cassandra should still be running.)

You can also install Cassandra in unattended mode, for example:

```
./APIGateway_7.6.2_Install_linux-x86-64_BN<n>.run --mode unattended
--setup_type advanced
--enable-components cassandra
--disable-components apigateway,qstart,policystudio,analytics,
configurationstudio,apitester,apimgmt,packagedeploytools
--cassandraInstalldir /opt/db/cassandra-2212
--cassandraJDK /opt/jre
--startCassandra 0
```

For more information on installing Cassandra, see *Install an Apache Cassandra database* in the *API Gateway Installation Guide*.

Step 2 - Run nodetool drain on Cassandra 2.2.8

Run the following commands to drain the node and flush the memTables to the SSTables before copying data to the Cassandra 2.2.12 installation. Writes are not accepted while this is happening.

```
$ cd /opt/db/cassandra-228/cassandra/bin
$ ./nodetool drain
```

Step 3 - Stop Cassandra 2.2.8

Run the following commands to stop Cassandra.

```
$ ps –ef | grep cassandra
$ sudo kill -9 <cassandra_pid>
```

For more information on stopping Cassandra on Linux, see *Manage Apache Cassandra* in the *API Gateway Apache Cassandra Administrator Guide*.

Step 4 - Copy data from Cassandra 2.2.8 to Cassandra 2.2.12

Run the following command to copy the data folder and all subfolders from your Cassandra 2.2.8 installation to your new Cassandra 2.2.12 installation.

```
$ cp -R /opt/db/cassandra-228/cassandra/data /opt/db/cassandra-2212/cassandra
```

For example, after running this command, you should have the following directories:

- `/opt/db/cassandra-2212/cassandra/data/commitlog`
- `/opt/db/cassandra-2212/cassandra/data/data`
- `/opt/db/cassandra-2212/cassandra/data/saved_caches`

Tip By default, all Cassandra data is stored in the `data` directory. However, in a production environment you should store the commit log (for example, `/opt/db/cassandra-2212/cassandra/data/commitlog`) on a separate disk partition, or a separate physical device from the data file directories. You can change the default locations in `cassandra.yaml` (`commitlog_directory`, `data_file_directories`, and `saved_caches_directory` properties). For more information, see http://docs.datastax.com/en/archived/cassandra/2.2/cassandra/configuration/configCassandra_yaml.html.

Step 5 - Copy SSL certificates from Cassandra 2.2.8 to Cassandra 2.2.12

If you have SSL certificates in your Cassandra 2.2.8 installation, copy them to Cassandra 2.2.12. To copy the SSL certificates, copy the following files to `/opt/db/cassandra-2212/cassandra/conf/`:

- `/opt/db/cassandra-228/cassandra/conf/.truststore`
- `/opt/db/cassandra-228/cassandra/conf/.keystore`

Step 6 - Update Cassandra 2.2.12 configuration files

Update your Cassandra 2.2.12 configuration files with the relevant settings from your Cassandra 2.2.8 installation. The following files must be updated:

- `/opt/db/cassandra-2212/cassandra/conf/cassandra.yaml`
- `/opt/db/cassandra-2212/cassandra/bin/cassandra.in.sh`

- /opt/db/cassandra-2212/cassandra/conf/cassandra.rackdc.properties
- /opt/db/cassandra-2212/cassandra/conf/cassandra-topology.properties
- /opt/db/cassandra-2212/cassandra/conf/cassandra-env.sh (This file only needs to be updated if you changed your JMX configuration after Cassandra 2.2.8 installation)

You can do a diff on the files to see a complete list of the differences. The following are the values in each file that must be updated:

/opt/db/cassandra-2212/cassandra/conf/cassandra.yaml:

```
rpc_address: <Set to the IP address of this Cassandra node>

listen_address: <Set to the IP address of this Cassandra node>

seed_provider:
    # Addresses of hosts that are deemed contact points.
    # Cassandra nodes use this list of hosts to find each other and learn
    # the topology of the ring.  You must change this if you are running
    # multiple nodes!
    - class_name: org.apache.cassandra.locator.SimpleSeedProvider
      parameters:
          # seeds is actually a comma-delimited list of addresses.
          # Ex: "<ip1>,<ip2>,<ip3>"
          - seeds: "<IP seed1>,<IP seed 2>"

server_encryption_options: <Set to same values as your 2.2.8 installation>

client_encryption_options: <Set to same values as your 2.2.8 installation>
```

/opt/db/cassandra-2212/cassandra/bin/cassandra.in.sh:

```
JAVA_HOME=<Set to the location of a 64-bit JRE, same value as your 2.2.8
installation>
```

/opt/db/cassandra-2212/cassandra/conf/cassandra.rackdc.properties:

Perform a diff on this file to identify the rack settings to update.

/opt/db/cassandra-2212/cassandra/conf/cassandra-topology.properties:

Perform a diff on this file to identify the rack settings to update.

/opt/db/cassandra-2212/cassandra/conf/cassandra-env.sh

If you changed your JMX configuration after Cassandra 2.2.8 installation, perform a diff on this file to identify the JMX settings to update.

Step 7 - Start Cassandra 2.2.12

Run the following commands to start Cassandra.

```
$ cd /opt/db/cassandra-2212/cassandra/bin
$ ./cassandra
```

For more information on starting Cassandra on Linux, see *Manage Apache Cassandra* in the *API Gateway Apache Cassandra Administrator Guide*.

Step 8 - Run nodetool upgradesstables on Cassandra 2.2.12

Run the following commands to rewrite SSTables that are not on the current version and upgrade them to Cassandra version 2.2.12.

```
$ cd /opt/db/cassandra-2212/cassandra/bin
$ ./nodetool upgradesstables
```

Step 9 - Run nodetool repair on Cassandra 2.2.12

Run the following commands to repair the tables.

```
$ cd /opt/db/cassandra-2212/cassandra/bin
$ ./nodetool repair
```

Cassandra upgrade steps - Multi-node single datacenter

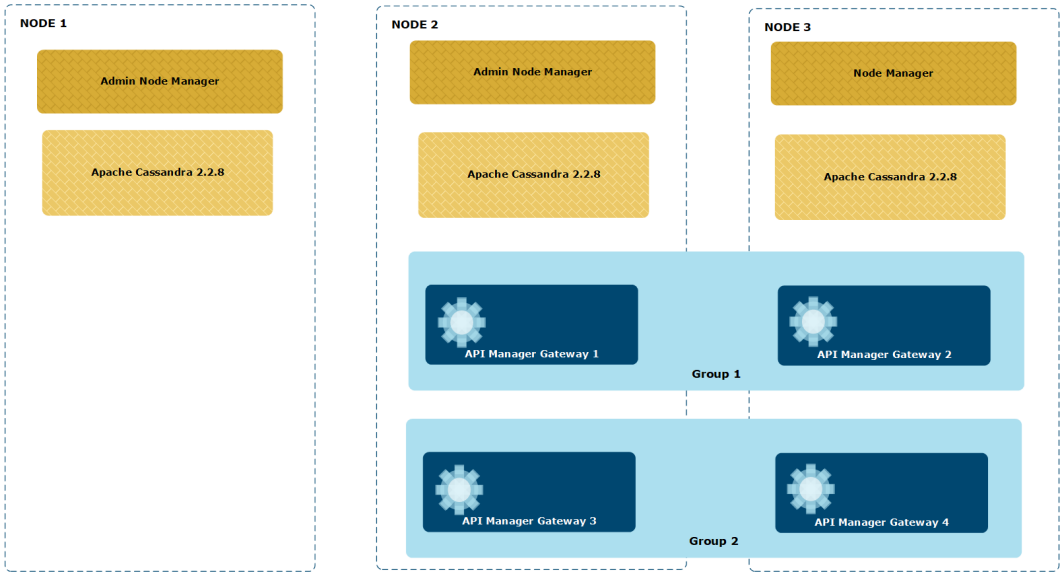
The following steps give an example of how to upgrade Cassandra in a single datacenter HA setup. In this example the datacenter has 2 API Gateways, 3 Cassandra nodes, and 2 groups.

The following steps give an example of how to upgrade a Cassandra cluster on a three-node HA setup with the following topology:

- Node 1 – Admin Node Manager, Cassandra 2.2.8
- Node 2 – Admin Node Manager, API Gateway, API Manager, Cassandra 2.2.8
- Node 3 – Node Manager, API Gateway, API Manager, Cassandra 2.2.8

There are two API Gateway groups, therefore two keyspaces. The groups are as follows:

- `Group1` has two API Manager-enabled API Gateway instances (one running on Node 2 and another on Node 3).
- `Group2` has two API Manager-enabled API Gateway instances (one running on Node 2 and another on Node 3).



Cassandra is set up as follows:

- Cassandra read/write consistency is set to Local Quorum
- Cassandra authentication is set (user name and password set to `cassUser`/`cassPasswd`)
- SSL encryption is enabled on Cassandra

Node 1

Perform the following steps on Node 1.

Step 1 - Install Cassandra 2.2.12 on Node 1

Follow these steps to install Apache Cassandra 2.2.12 using the API Gateway installer in default GUI mode.

1. Select the **Custom** option in the installer.
2. When prompted to select components to install, select only the **Cassandra** component.
3. When prompted for the Cassandra configuration, enter the following settings:

Installation Directory	JRE Location
`/opt/db/cassandra-2212`	`/opt/jre`

4. Do not start Cassandra 2.2.12 when the installation completes. (Your earlier version of Cassandra should still be running.)

You can also install Cassandra in unattended mode, for example:

```
./APIGateway_7.6.2_Install_linux-x86-64_BN<n>.run --mode unattended
--setup_type advanced
--enable-components cassandra
--disable-components apigateway,qstart,policystudio,analytics,
configurationstudio,apitester,apimgmt,packagedeploytools
--cassandraInstalldir /opt/db/cassandra-2212
--cassandraJDK /opt/jre
--startCassandra 0
```

For more information on installing Cassandra, see *Install an Apache Cassandra database* in the *API Gateway Installation Guide*.

Step 2 - Run nodetool drain on Cassandra 2.2.8

Run the following commands to drain the node and flush the memTables to the SSTables before copying data to the Cassandra 2.2.12 installation. Writes are not accepted while this is happening.

```
$ cd /opt/db/cassandra-228/cassandra/bin
$ ./nodetool drain
```

Step 3 - Stop Cassandra 2.2.8

Run the following commands to stop Cassandra.

```
$ ps –ef | grep cassandra
$ sudo kill -9 <cassandra_pid>
```

For more information on stopping Cassandra on Linux, see *Manage Apache Cassandra* in the *API Gateway Apache Cassandra Administrator Guide*.

Step 4 - Copy data from Cassandra 2.2.8 to Cassandra 2.2.12

Run the following command to copy the data folder and all subfolders from your Cassandra 2.2.8 installation to your new Cassandra 2.2.12 installation.

```
$ cp –R /opt/db/cassandra-228/cassandra/data /opt/db/cassandra-2212/cassandra
```

For example, after running this command, you should have the following directories:

- `/opt/db/cassandra-2212/cassandra/data/commitlog`
- `/opt/db/cassandra-2212/cassandra/data/data`

- /opt/db/cassandra-2212/cassandra/data/saved_caches

Tip By default, all Cassandra data is stored in the data directory. However, in a production environment you should store the commit log (for example, /opt/db/cassandra-2212/cassandra/data/commitlog) on a separate disk partition, or a separate physical device from the data file directories. You can change the default locations in cassandra.yaml (commitlog_directory, data_file_directories, and saved_caches_directory properties). For more information, see http://docs.datastax.com/en/archived/cassandra/2.2/cassandra/configuration/configCassandra_yaml.html.

Step 5 - Copy SSL certificates from Cassandra 2.2.8 to Cassandra 2.2.12

If you have SSL certificates in your Cassandra 2.2.8 installation, copy them to Cassandra 2.2.12. To copy the SSL certificates, copy the following files to /opt/db/cassandra-2212/cassandra/conf/:

- /opt/db/cassandra-228/cassandra/conf/.truststore
- /opt/db/cassandra-228/cassandra/conf/.keystore

Step 6 - Update Cassandra 2.2.12 configuration files

Update your Cassandra 2.2.12 configuration files with the relevant settings from your Cassandra 2.2.8 installation. The following files must be updated:

- /opt/db/cassandra-2212/cassandra/conf/cassandra.yaml
- /opt/db/cassandra-2212/cassandra/bin/cassandra.in.sh
- /opt/db/cassandra-2212/cassandra/conf/cassandra.rackdc.properties
- /opt/db/cassandra-2212/cassandra/conf/cassandra-topology.properties
- /opt/db/cassandra-2212/cassandra/conf/cassandra-env.sh (This file only needs to be updated if you changed your JMX configuration after Cassandra 2.2.8 installation)

You can do a diff on the files to see a complete list of the differences. The following are the values in each file that must be updated:

/opt/db/cassandra-2212/cassandra/conf/cassandra.yaml:

```
rpc_address: <Set to the IP address of this Cassandra node>

listen_address: <Set to the IP address of this Cassandra node>

seed_provider:
    # Addresses of hosts that are deemed contact points.
```

```
    # Cassandra nodes use this list of hosts to find each other and learn
    # the topology of the ring.  You must change this if you are running
    # multiple nodes!
    - class_name: org.apache.cassandra.locator.SimpleSeedProvider
      parameters:
          # seeds is actually a comma-delimited list of addresses.
          # Ex: "<ip1>,<ip2>,<ip3>"
          - seeds: "<IP seed1>,<IP seed 2>"

server_encryption_options: <Set to same values as your 2.2.8 installation>

client_encryption_options: <Set to same values as your 2.2.8 installation>
```

`/opt/db/cassandra-2212/cassandra/bin/cassandra.in.sh`:

```
JAVA_HOME=<Set to the location of a 64-bit JRE, same value as your 2.2.8
installation>
```

`/opt/db/cassandra-2212/cassandra/conf/cassandra.rackdc.properties`:

Perform a diff on this file to identify the rack settings to update.

`/opt/db/cassandra-2212/cassandra/conf/cassandra-topology.properties`:

Perform a diff on this file to identify the rack settings to update.

`/opt/db/cassandra-2212/cassandra/conf/cassandra-env.sh`

If you changed your JMX configuration after Cassandra 2.2.8 installation, perform a diff on this file to identify the JMX settings to update.

Step 7 - Start Cassandra 2.2.12

Run the following commands to start Cassandra.

```
$ cd /opt/db/cassandra-2212/cassandra/bin
$ ./cassandra
```

For more information on starting Cassandra on Linux, see *Manage Apache Cassandra* in the *API Gateway Apache Cassandra Administrator Guide*.

Step 8 - Run nodetool upgradesstables on Cassandra 2.2.12

Run the following commands to rewrite SSTables that are not on the current version and upgrade them to Cassandra version 2.2.12.

```
$ cd /opt/db/cassandra-2212/cassandra/bin
$ ./nodetool upgradesstables
```

Node 2

Repeat steps 1 to 8 on Node 2.

Node 3

Repeat steps 1 to 8 on Node 3.

Final step - Run nodetool repair on Cassandra 2.2.12 on each node

Run the following commands, one at a time on each node, to repair the tables.

```
$ cd /opt/db/cassandra-2212/cassandra/bin
$ ./nodetool repair
```

Cassandra upgrade - Multi-datacenter

To upgrade Apache Cassandra in a multi-DC setup, you can follow the same steps as for the multi-node single-DC setup:

- Repeat the steps on each node in the cluster
- Upgrade all of the nodes in one DC before moving to the next DC

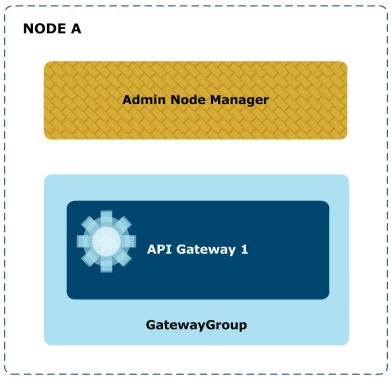
Single-node upgrade example (upgrades from 7.5.x or 7.6.x)

This topic provides an example of a single-node domain upgrade from API Gateway version 7.5.x or 7.6.x (in this case, 7.5.1) to API Gateway7.6.2.

Tip You can use the steps in this example as a guide when upgrading a single-node domain from API Gateway 7.5.x or 7.6.x to 7.6.2. However, you must remember to modify the steps appropriately for your version and topology.

Sample upgrade topology

The sample topology used in this example is as follows:



Summary of steps

The steps required to perform a single-node upgrade from API Gateway 7.5.1 to API Gateway 7.6.2 are similar to those required for a single-node upgrade from 7.3.x or 7.4.x versions as detailed in *Single-node upgrade example (upgrades from 7.3.x or 7.4.x)* on page 26, with a few significant differences. This section summarizes the steps and highlights the differences:

1. Perform the checks on your old API Gateway 7.5.1 installation, as detailed in *Checklist for the old API Gateway installation* on page 42.
    - No Apache Cassandra checks are necessary on the old installation, as the external Cassandra configuration is retained when upgrading from 7.5.x or 7.6.x.
2. Install API Gateway7.6.2. Perform the same steps detailed in *Step 2 – Install API Gateway 7.6.2* on page 27 with the following differences:
    - Do not select Cassandra in the **Custom** installation.
3. When the installation is complete, perform the new installation checks detailed in *Checklist for the new API Gateway 7.6.2 installation* on page 44.
    - You do not need to configure an Apache Cassandra database cluster in the new installation, as the external Cassandra configuration is retained when upgrading from 7.5.x or 7.6.x.
    - You must open the port 9042 on your firewall to enable API Gateway to communicate with Apache Cassandra as described in *Open the new Apache Cassandra client port in the firewall* on page 44. Alternatively, you can configure API Gateway to use a different port after upgrade as described in *Configure a different Apache Cassandra client port* on page 58.
4. Run the `export` and `upgrade` commands. Perform the same steps detailed in *Step 4 – Run export and upgrade commands* on page 27 with the following differences:
    - If you are using Apache Cassandra, you do not need to specify the host name or IP address of the existing external Cassandra database cluster to the `upgrade` command (the Cassandra configuration is retained when upgrading from 7.5.x or 7.6.x).
    - If you are not using Apache Cassandra, you do not need to run the `upgrade` command with the `--no_cassandra` option

The following example shows how to run the `export` and `upgrade` commands:

```
> cd /opt/Axway-7.6.2/apigateway/upgrade/bin
> ./sysupgrade export --old_install_dir /opt/Axway-7.5.1/apigateway/
> ./sysupgrade upgrade
```

5. Run the `apply` command. Perform the same steps detailed in *Step 5 – Run apply command* on page 28 with the following differences:
   - You do not need to update the `cassandra.yaml` file or start Cassandra, as Cassandra should already be running.

   The following example shows how to run the `apply` command:

```
> cd /opt/Axway-7.6.2/apigateway/upgrade/bin
> ./sysupgrade apply
```

6. Verify the upgrade as detailed in *Step 6 – Verify the upgrade* on page 29.

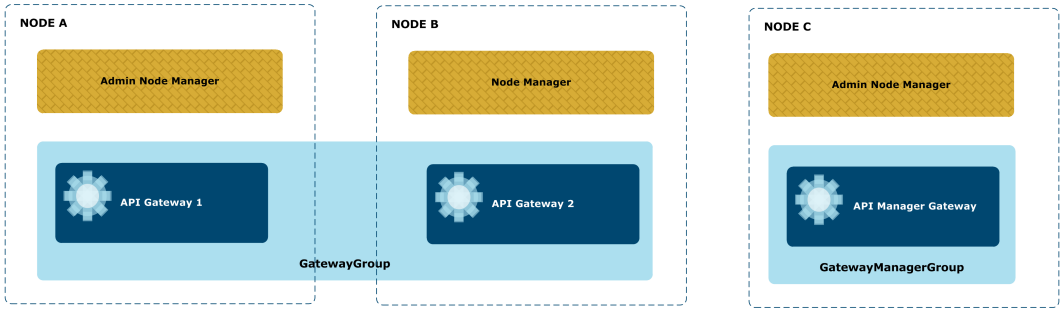
Multi-node upgrade example (upgrades from 7.5.x or 7.6.x)

This topic provides an example of a multi-node domain upgrade from API Gateway version 7.5.x or 7.6.x (in this case, 7.5.1) to API Gateway 7.6.2.

Tip You can use the steps in this example as a guide when upgrading a multi-node domain from API Gateway 7.5.x or 7.6.x to 7.6.2. However, you must remember to modify the steps appropriately for your version and topology.

Sample upgrade topology

The sample topology used in this example is as follows:



Summary of steps

The steps required to perform a multi-node upgrade from API Gateway 7.5.1 to API Gateway 7.6.2 are similar to those required for a multi-node upgrade from 7.3.x or 7.4.x versions as detailed in *Multi-node upgrade example (upgrades from 7.3.x or 7.4.x)* on page 30, with a few significant differences. This section summarizes the steps and highlights the differences:

1. Perform the checks on your old API Gateway 7.5.1 installation, as detailed in *Checklist for the old API Gateway installation* on page 42.
    - No Apache Cassandra checks are necessary on the old installation, as the external Cassandra configuration is retained when upgrading from 7.5.x or 7.6.x.
2. Install API Gateway 7.6.2 on each node (NodeA, NodeB, and NodeC). Perform the same steps detailed in *Step 2 – Install API Gateway 7.6.2 on each node* on page 32 with the following differences:
    - Do not select Cassandra in the **Custom** installation.
3. When the installation is complete, perform the new installation checks detailed in *Checklist for the new API Gateway 7.6.2 installation* on page 44.
    - You do not need to configure an Apache Cassandra database cluster in the new installation, as the external Cassandra configuration is retained when upgrading from 7.5.x or 7.6.x.
    - You must open the port 9042 on your firewall to enable API Gateway to communicate with Apache Cassandra as described in *Open the new Apache Cassandra client port in the firewall* on page 44. Alternatively, you can configure API Gateway to use a different port after upgrade as described in *Configure a different Apache Cassandra client port* on page 58.
4. Run the `export` and `upgrade` commands on each node (NodeA, NodeB, and NodeC). Perform the same steps detailed in *Step 4 – Run export and upgrade on each node* on page 33 with the following differences:
    - If you are using Apache Cassandra, you must still specify the host name or IP address of the existing external Cassandra database cluster to the `upgrade` command using the `--cass_host` option. This is necessary for multi-node upgrades from 7.5.x or 7.6.x to enable all API Gateways to communicate with the external Cassandra server.
    - If you are not using Apache Cassandra, you must still run the `upgrade` command with the `--no_cassandra` option.

    The following example shows how to run the `export` and `upgrade` commands on each node:

```
> cd /opt/Axway-7.6.2/apigateway/upgrade/bin

> ./sysupgrade export --old_install_dir /opt/Axway-7.5.1/apigateway/ --anm_host
NodeA

> ./sysupgrade upgrade --cass_host NodeA
```

5. Run `apply` on the first Admin Node Manager node (NodeA). Perform the same steps detailed in *Step 5 – Run apply on the first Admin Node Manager (NodeA)* on page 35 with the following differences:
   - You do not need to update the `cassandra.yaml` file or start Cassandra, as Cassandra should already be running.

   The following example shows how to run the `apply` command on NodeA:

```
> cd /opt/Axway-7.6.2/apigateway/upgrade/bin
> ./sysupgrade apply --anm_host NodeA
```

6. Run `apply` on the other nodes in turn (NodeB and then NodeC). Perform the same steps as detailed in *Step 6 – Run apply on the other nodes* on page 36.

   The following example shows how to run the `apply` command on each of the other nodes:

```
> cd /opt/Axway-7.6.2/apigateway/upgrade/bin
> ./sysupgrade apply --anm_host NodeA
```

7. Verify the upgrade as detailed in *Step 7 – Verify the upgrade* on page 37.

After you upgrade from 7.5.x or 7.6.x

This topic includes post-upgrade steps that you might need to perform after running `sysupgrade` to upgrade from API Gateway 7.5.x or 7.6.x to 7.6.2. It contains the following topics:

- *Configure a different Apache Cassandra client port* on page 58
- *Upgrade API Gateway projects* on page 59
- *Upgrade services* on page 59
- *Migrate the QuickStart tutorial* on page 59

Configure a different Apache Cassandra client port

If you upgraded from API Gateway 7.5.1 or later to version 7.6.2 all Cassandra hosts are updated to use port 9042 for client communication. To use a different port (for example, to use the same port as you used in your old installation), follow these steps:

1. For each Cassandra host, update the setting `native_transport_port` in the `CASSANDRA_HOME/conf/cassandra.yaml` file. Set the value to the port number to use for client communication.
2. Update the details for each Cassandra host in Policy Studio. Select **Server Settings > Cassandra > Hosts**, and update the port for each host.

Note If you change the configuration to use the same port as you used in your old installation, you cannot leave any API Gateways in your old installation running during the `apply` step, as the port uses a different Cassandra protocol.

Upgrade API Gateway projects

Each API Gateway group has a configuration that is typically deployed as a `.fed` file. When you upgrade from an earlier version of API Gateway, configuration for all API Gateway groups is automatically upgraded during `sysupgrade`. However, you might have configuration files that were originally created in Policy Studio in a development environment that also need to be upgraded. You can upgrade the configuration in your development environment in one of the following ways:

- In Policy Studio:
    - Choose the **From an API Gateway instance** option to create a new project from the configuration in an already upgraded API Gateway.
    - Choose the **From existing configuration** option to create a new project from an old configuration. The configuration is upgraded to version 7.6.2 automatically.

  For more information on creating projects in Policy Studio, see the *API Gateway Policy Developer Guide*.
- If you upgraded from version 7.5.1 or later and you have several projects to upgrade (these projects might be independent of one another, or could include shared projects and their dependencies), you can use the `projupgrade` tool. This tool upgrades several projects at once. For more information, see "Upgrade an API Gateway project" in the *API Gateway DevOps Deployment Guide*.

Upgrade services

If you were running the API Gateway and Node Manager processes as services in your old installation, you must update the service scripts manually after the upgrade completes. Follow the steps in *Upgrade services* on page 38.

Migrate the QuickStart tutorial

`sysupgrade` does not migrate the Quickstart tutorial from your old installation. To migrate it, copy the `/apigateway/webapps/quickstart` directory from your old installation (for example, `/opt/Axway/7.4.1/apigateway/webapps/quickstart`) to the same location in the new 7.6.2 installation (for example, `/opt/Axway/7.6.2/apigateway/webapps/quickstart`).

Zero downtime upgrade

The standard process to upgrade to API Gateway 7.6.2 involves a short period of downtime during the apply phase. With most `.fed` files, this downtime should not be more than a few minutes. However, with bigger `.fed` files of large and complex configurations, this downtime can be longer. This topic describes an approach you can take, and some sample scripts you can use as a reference, to achieve a zero downtime upgrade (ZDU) to API Gateway 7.6.2.

This approach involves the use of a load balancer to ensure that available API Gateways can always process traffic, and if you are using a DevOps framework, the ZDU sample scripts provide an example for a basic high availability (HA) deployment and an nginx load balancer, to help you understand the required steps. The ZDU sample scripts provide an example only, and although the scripts are somewhat configurable, you must adapt them for your specific needs.

The ZDU scripts package is available from Axway Support at https://support.axway.com. The package includes scripts for Linux.

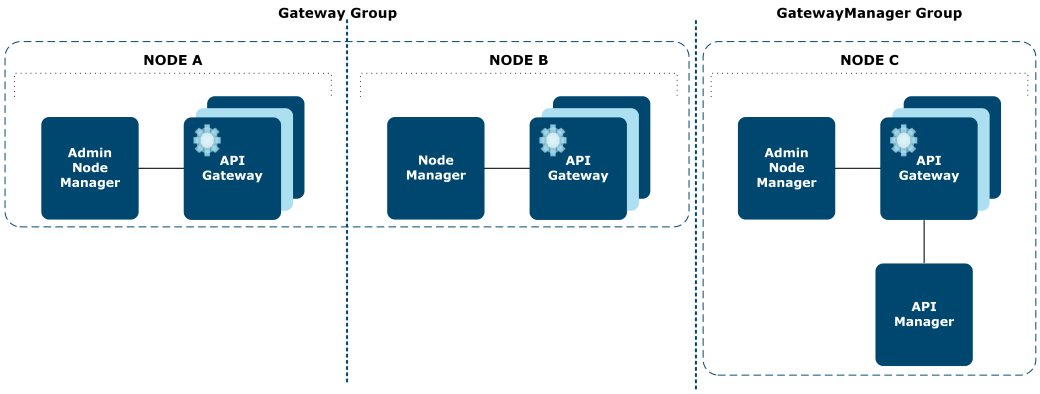
Note

- Use the ZDU sample scripts only for upgrading from API Gateway 7.5.2 or later to 7.6.2 when Cassandra contains all the shared data.
- We recommend that you perform a full upgrade that is completed in a single attempt. Because shared data (in particular quota counts) are involved, it is possible that this data could degrade if an upgraded subset of the domain coexists with a non-upgraded subset for a significant period of time.

Reference configuration

The reference configuration is a three-node topology configured as follows:

- Two Admin Node Managers, one on Node A and the other on Node C.
- One Node Manager on Node B.
- Each node runs a single API Gateway instance that are grouped as follows:
  - `Gateway Group` containing API Gateway instances on Node A and Node B.
  - `GatewayManager Group` containing a single API Gateway instance running API Manager.



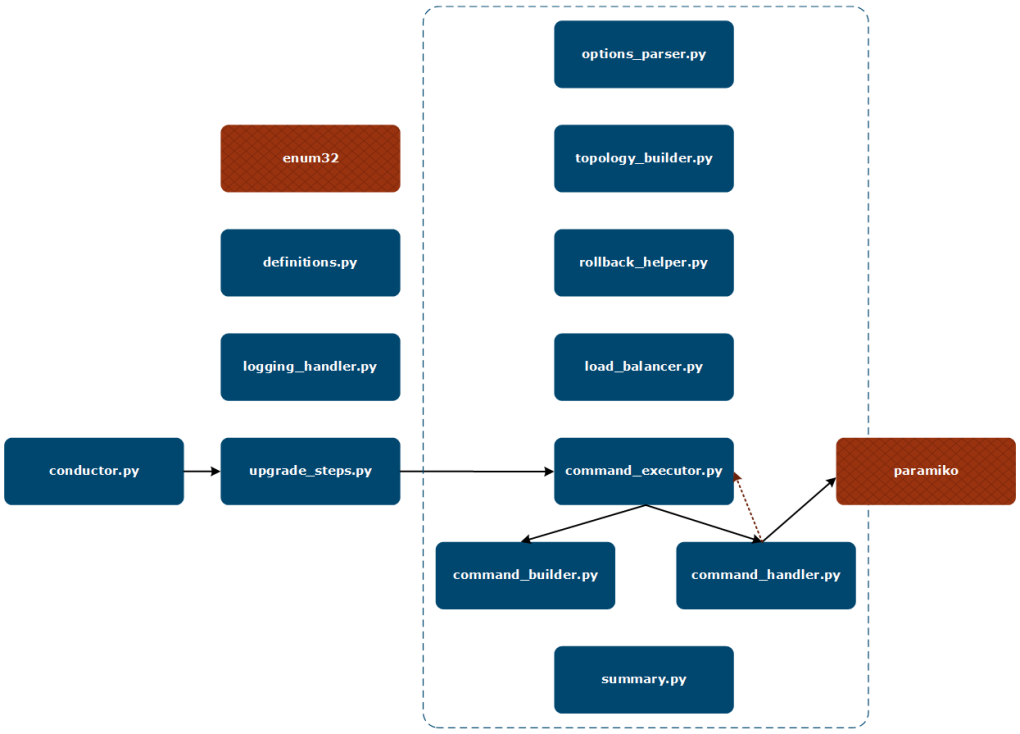
ZDU script package

The ZDU script package (for example, `APIGateway_7.6.2_Package_ZDUScripts_linux-x86-64_BN`*`YYYYMMDDn`*`.zip`) contains the following folders and files:

- `README.md`: This file explains how the ZDU scripts work.
- `bin`: This folder contains the `zdupgrade` shell script for use on Linux.
- `config`: This folder contains a `topology.json` file for a sample configuration.
- `scripts`: This folder contains the individual sample scripts that the `zdupgrade` script uses. You can adapt these scripts to your specific needs.

Sample scripts

The following diagram shows the architecture.



The following table describes the purpose of each sample script.

Script	Description
`command_builder.py`	Generates the Linux commands to be run (for example, `sysupgrade` commands, commands to start or stop processes).
`command_executor.py`	Executes commands on a node (for example, run a `sysupgrade` step, start or stop a process, check if a process is running, roll back). Requires `command_builder` and `command_handler`.
`command_handler.py`	Handles how commands are run on a node (for example, how to connect to remote nodes to run commands).
`conductor.py`	Orchestrates the zero downtime upgrade process (for example, connect to all nodes, run sysupgrade steps on each node in sequence, disconnect from all nodes, display a summary).
`definitions.py`	Defines variables used throughout the other scripts.
`load_balancer.py`	Placeholder script to implement load balancer logic (for example, to enable or disable traffic on a node).

Script	Description
`logging_handler.py`	Provides logging capabilities.
`options_parser.py`	Parses and validates command-line options.
`rollback_helper.py`	Analyzes errors generated during the ZDU process and determines if a roll-back is needed.
`summary.py`	Provides a summary of ZDU results per node and per upgrade phase.
`topology_builder.py`	Builds the topology information from the configuration file.
`upgrade_steps.py`	Implements the ZDU process steps on each node.

See the comments included in each script for more detailed information. For more information on using and running the scripts, see *Use the ZDU scripts* on page 63.

Use the ZDU scripts

This topic describes how to use the sample ZDU scripts. It contains the following sections:

- *Prerequisites* on page 63
- *Run the zdupgrade script* on page 64
- *Customize the zdupgrade sample scripts* on page 68
- *Troubleshooting* on page 69

Prerequisites

The `zdupgrade` scripts are intended to be run on a control node (for example, a local machine). This control node is responsible for connecting to and upgrading each of the nodes in the topology (remote nodes) in sequence.

The prerequisites for the node on which you intend to run the `zdupgrade` scripts (the control node) are as follows:

- You must use the same operating system on the control node and the remote nodes.
- You must install Python 2.6 or later (Python 2.7 recommended) and the `pip` package manager. For more details, see the Python documentation. We recommend that you do not install Python under a path that contains spaces.

- On Linux, you must install cryptography dependencies. For more details, see the Cryptography.io.
- You must install the `paramiko` and `enum34` Python modules and their dependencies.
- You must download and install the ZDU sample script package from Axway Support at https://support.axway.com.
- You must have an RSA key for password-less login to the remote nodes over SSH. For more information, see *Configure SSH server* on page 64.

The prerequisites for the remote nodes are as follows:

- You must install API Gateway version 7.6.2 on each node.

  Note The scripts expect the old installation location to be the same on each node and the new installation location to be the same on each node.

- You must have all required licenses for the API Gateway components on each node.
- The old installation Node Managers and API Gateways must be running.
- You must configure a SSH server on all nodes. For more information, see *Configure SSH server* on page 64.

Configure SSH server

To enable a user to log in to a remote node without a password, you must configure a SSH server on each remote node.

Additionally, follow these steps:

1. Use `ssh-keygen` to generate a RSA key for the SSH user, for example:

   ```
   ssh-keygen -t rsa
   ```

   Alternatively, if you already have an existing SSH key, you can convert it to an RSA key, for example:

   ```
   ssh-keyscan -t rsa localhost
   ```

2. Copy the RSA key to each of the remote nodes, for example:

   ```
   ssh-copy-id -i <ssh user home>/.ssh/id_rsa.pub <ssh user>@<node>
   ```

3. Ensure that the RSA key is the first in the list of known hosts for each remote node.

For more information on using OpenSSH, go to https://www.openssh.com/.

Run the zdupgrade script

This section describes how to install and run the `zdupgrade` script. It also describes the log files produced by the script.

Install the package

To install the package, unzip the package you downloaded from Axway Support at https://support.axway.com to a directory on your local machine (for example, `/opt/zdu`). For more information on the contents of the package, see *ZDU script package* on page 61.

Run with default options

To run the `zdupgrade` script with the default options, enter the following commands:

```
> cd opt/zdu/bin
> ./zdupgrade --old_install_dir OLDINSTALLDIR --new_install_dir NEWINSTALLDIR
```

- `OLDINSTALLDIR` is the location of the old API Gateway installation.
- `NEWINSTALLDIR` is the location of the new API Gateway 7.6.2 installation.

Note

- The scripts expect the old installation location to be the same on each node and the new installation location to be the same on each node.
- The `--old_install_dir` and `--new_install_dir` options are case sensitive.
- The scripts use the reference topology unless you specify a topology file using the `--config` option. A sample topology file is provided in `config/topology.json`, however you must update this file to match your configuration. Do not use the `topology.json` file in your API Gateway installation instead of this file as they have a different purpose and format. For more information, see *Specify topology configuration to use* on page 68.

View script options

To see the mandatory and optional arguments for the `zdupgrade` script, enter the following commands:

```
> cd opt/zdu/bin
> ./zdupgrade --help
```

Log files

When you run the `zdupgrade` script, it generates two types of log files :

- ZDU script log
- Node-specific log

The ZDU script log is a timestamped log file in the `/zdu/logs` directory. This log file tracks the progress of the `zdupgrade` script steps, and in case of a failure, logs the error message on what went wrong so you can fix any issues in the upgrade. This log file also contains a summary table for the upgrade, which you can use to quickly identify if the upgrade was successful, or if it was not successful you can see on which node and at what step the upgrade failed.

The following shows an example summary for a successful upgrade:

```
Summary table (Status.SUCCESS):

    +------------+------------+------------+------------+------------+------------+-
-----------+
    | MULTI-NODE ZERO DOWN-TIME UPGRADE SUMMARY
          |
    +------------+------------+------------+------------+------------+------------+-
-----------+
    | NODE       | EXPORT     | UPGRADE    | PRE-APPLY  | APPLY      | POST-APPLY |
ROLLBACK   |
    +------------+------------+------------+------------+------------+------------+-
-----------+
    | Node-A     | PASSED (0) | PASSED (0) | PASSED (0) | PASSED (0) | PASSED (0) |
-          |
    +------------+------------+------------+------------+------------+------------+-
-----------+
    | Node-B     | PASSED (0) | PASSED (0) | PASSED (0) | PASSED (0) | PASSED (0) |
-          |
    +------------+------------+------------+------------+------------+------------+-
-----------+
    | Node-C     | PASSED (0) | PASSED (0) | PASSED (0) | PASSED (0) | PASSED (0) |
-          |
    +------------+------------+------------+------------+------------+------------+-
-----------+
```

The following shows an example summary for a failed upgrade:

```
Summary table (Status.UPGRADE_FAILED):

    +--------------+--------------+--------------+--------------+--------------+----
----------+--------------+
    | MULTI-NODE ZERO DOWN-TIME UPGRADE SUMMARY
                        |
    +--------------+--------------+--------------+--------------+--------------+----
----------+--------------+
    | NODE         | EXPORT       | UPGRADE      | PRE-APPLY    | APPLY        |
POST-APPLY   | ROLLBACK     |
    +--------------+--------------+--------------+--------------+--------------+----
----------+--------------+
    | Node-A       | PASSED (0)   | PASSED (0)   | -            | -            | -
         | -            |
    +--------------+--------------+--------------+--------------+--------------+----
----------+--------------+
    | Node-B       | PASSED (0)   | PASSED (0)   | -            | -            | -
```

```
         | -            |
    +--------------+--------------+--------------+--------------+--------------+----
----------+--------------+
    | Node-C       | PASSED (0)   | FAILED (100) | -            | -            | -
         | -            |
    +--------------+--------------+--------------+--------------+--------------+----
----------+--------------+
```

A return status code is included in the summary (for example, `Status.UPGRADE_FAILED` indicates that the `sysupgrade upgrade` step failed). The full list of return status codes are explained in the comments of the `definitions.py` script but are reproduced here for convenience.

Return Status Code	Value
SUCCESS	0
EXPORT_FAILED	70
UPGRADE_FAILED	71
PRE_APPLY_FAILED	72
APPLY_FAILED	73
POST_APPLY_FAILED	74
ROLLBACK_FAILED	75
FATAL_ERROR	80

The node-specific logs are in the `/zdu/logs/<node name>` directory. The node-specific logs contain the output and possible errors for the commands run on that particular node. When troubleshooting a failure, you can check the node-specific log file to identify the exact source of the error.

For example, for the upgrade failure on Node-C above, the Node-C log file shows the following:

```
########## SYSUPGRADE_UPGRADE ##########
------------- ZDU_EXEC_CMD -------------
command [
    cd /opt/Axway-7.5.3/apigateway/upgrade/bin
    /opt/Axway-7.5.3/apigateway/upgrade/bin/sysupgrade upgrade --cass_host localhost
--cass_port 9160
]
ERROR: This is a multi-node upgrade and a localhost Cassandra server: localhost:9160
has been specified.
You must use a network accessible address. Run sysupgrade upgrade -h for details.
```

```
------------- ZDU_EXEC_CMD -------------
########## SYSUPGRADE_UPGRADE ##########
```

Customize the zdupgrade sample scripts

This section provides some examples of how you might customize the sample scripts for your own use, and provides suggested steps for modifying the sample scripts.

Perform the following steps to modify the sample scripts:

1. Change to the `zdu/scripts` directory.
2. Open the sample script you want to change in your preferred editor.
3. Edit the script to suit your configuration. You can also enable or disable the script as needed.
4. Save the modified script.
5. Change to the `zdu/bin` directory, and run the `zdupgrade` script with the desired options.
6. Change to the `zdu/logs` directory and check the results. If further adjustments to the script are needed, repeat steps 2 to 6.

Specify topology configuration to use

You must specify your topology configuration to the `zdupgrade` script using the `--config <path_to_the_user_topology_file>` option.

If you do not specify any topology configuration using the `--config` option, the default topology file (`config/topology.json`) is not used, and the reference configuration is used (see *Reference configuration* on page 60).

Alternatively, you can implement the method `__getTopology(anmHost)` in `topology_builder.py` to return the correct topology configuration for your environment.

Note The intention of the `__getTopology` method is to autodiscover the topology from the Admin Node Manager host, meaning that a topology configuration file would not be required.

Escape argument values

The argument values are not escaped before they are used to build the upgrade commands to be executed in the nodes. If required, you must modify the `command_builder.py` script to escape the values accordingly. For example, on Linux, you could place values containing spaces in quotes.

Customize roll-back operation

If the upgrade fails at any stage for any reason, the `zdupgrade` script can perform a roll-back operation and return the configuration to the state it was before running the `zdupgrade` script.

Use the `rollback_helper.py` script to decide if a roll-back is required depending on the upgrade phase and its result. The results for the previously executed upgrade phases are also available from the `node.result` parameter.

You must resolve and fix any failures during the roll-back procedure based on the feedback from the ZDU scripts and logs.

You can customize the error handling routine in the script to ignore some errors and let the upgrade proceed to the next node. In this case, you must fix the errors in the failing nodes and take care of reconfiguring the new system to match the old configuration if required.

Enable or disable traffic

The `load_balancer.py` script is a placeholder to enable or disable traffic to a node through a load balancer if required. You must implement the `enable` and `disable` methods in this script to suit your own load balancing solution.

This script is already called as part of the ZDU processing with a dummy implementation.

Customize startup and shutdown times

You can customize the variables that control the time to wait for processes to start or stop, depending on how long it takes your system to start up or shut down. These variables are set in `command_executor.py`. The defaults in seconds are:

```
STARTUP_TIME = 30
SHUTDOWN_TIME = 10
```

Troubleshooting

This section details some common problems that you might encounter when running the `zdupgrade` scripts, and provides suggested workarounds.

Script fails if the Python path contains spaces

The `zdupgrade` scripts do not expect to have spaces in the Python path. For example, the script fails if Python is installed in `/opt/My Python/bin/`.

We recommend that you do not install Python under a path that contains spaces.

Alternatively, you can change the scripts to surround any variables with quotes to avoid the error.

Upgrade API Gateway Analytics 5

If you are using an earlier version of API Gateway Analytics, you must follow the steps in this section to upgrade API Gateway Analytics to version 7.6.2. Data in your old API Gateway Analytics metrics database will be preserved and the upgraded database will be fully compatible with version 7.6.2.

Note For details on upgrading a metrics database for use with API Manager, see *Upgrade your metrics database for API Manager* on page 81 instead.

You can upgrade API Gateway Analytics before or after you upgrade API Gateway. This depends on the version of API Gateway Analytics you are upgrading from:

- If you are upgrading from 7.4.0 and later versions, we recommend that you upgrade API Gateway Analytics before upgrading API Gateway using the `sysupgrade` command.

  If you create a new API Gateway Analytics metrics database as part of a rollback strategy, you must run `managedomain` on every host to change the database URL to that of the newly created database after you have completed the `sysupgrade apply` step.
- If you are upgrading from versions earlier than 7.4.0, we recommend that you upgrade API Gateway Analytics after upgrading API Gateway using the `sysupgrade` command.

For frequently asked questions about upgrading API Gateway Analytics, see *API Gateway Analytics and metrics database upgrades* on page 118.

Summary of steps

The following summarizes the steps to upgrade API Gateway Analytics. Some of the steps are optional, depending on what version you are upgrading from.

1. Install API Gateway Analytics7.6.2 to a new directory.
2. Copy any third-party JDBC drivers to the new installation.
3. Back up the old API Gateway Analytics metrics database.
4. If your version of API Gateway Analytics is earlier than 7.4.0, run `dbsetup` to upgrade your metrics database.
5. Run `upgradeconfig` to migrate your old API Gateway Analytics Entity Store customizations to the new installation.
6. Run `configureserver` to configure your new API Gateway Analytics Entity Store.
7. Migrate any custom reports.
8. Stop the old version of the API Gateway Analytics service and start the new 7.6.2 version.

9. If you are not upgrading from 7.4.0, 7.4.1, or 7.4.2 with metrics already enabled, run `managedomain` to enable API Gateway Analytics for the Node Managers. You must also run `managedomain` to update the metrics database URL if you created a new database (for example, as part of a rollback strategy).

Rollback strategy

If you want to be able to revert back to your old version of API Gateway Analytics and API Gateway, the best approach is to create a new API Gateway Analytics metrics database for version 7.6.2. The old versions can then be relaunched without changes. Where appropriate, this section details additional tasks that you need to perform to implement this rollback strategy.

Step 1 - Install API Gateway Analytics 7.6.2

You can install API Gateway Analytics on the same machine as your old API Gateway Analytics, or on a different machine. If you install it on the same machine as your old installation, you must install it in a new directory (and not in the same directory as the old installation). You do not need to install API Gateway Analytics on the same machine as any running API Gateways, therefore, you can install it on a dedicated machine if required.

Rollback strategy

To be able to rollback, you must create a new database with a different name to the old database, so that your old database remains unmodified. In the following sections, the new database is called `ReporterNew`.

If you are not implementing a rollback strategy, and your old database is earlier than version 7.4.0, you must upgrade your database as detailed in *Step 4 – Run dbsetup to upgrade the database* on page 72. You must also back up your old database as detailed in *Step 3 – Back up the database in the old installation* on page 72.

Step 2 - Copy third-party JDBC drivers to the new installation

You must copy the JDBC driver files for your chosen database from your old installation to your new installation of API Gateway, API Gateway Analytics, and Policy Studio.

For more information on where to copy the JDBC driver files from, see the section on configuring the database for API Gateway Analytics in the *API Gateway Installation Guide*. Copy them to the same locations in your new installation.

Step 3 - Back up the database in the old installation

You must backup the API Gateway Analytics database being used in the old installation before upgrading. For more details on backing up your database, see your database user documentation.

Rollback strategy

To rollback, you must perform the relevant SQL Restore operation to overwrite the newly created blank database (for example, `ReporterNew`) with the data and schema from the old API Gateway Analytics database.

Step 4 - Run dbsetup to upgrade the database

Note If you are upgrading from API Gateway Analytics version 7.4.0 or later, you do not need to run `dbsetup` to upgrade the database as it is already compatible with version 7.6.2.

You can use the utility `dbsetup` to upgrade the API Gateway Analytics database schema and data. This enables you to retain your old database data, but upgrade your database so that it is compatible with API Gateway Analytics7.6.2.The `dbsetup` utility always checks the existing version, and modifies the database only if an update is required.

The `dbsetup` utility is located in:

Windows

```
INSTALL_DIR\analytics\Win32\bin
```

Linux

```
INSTALL_DIR/analytics/posix/bin
```

To upgrade the database, follow these steps:

1. Stop any API Gateway processes or services in the old installation. This is necessary to prevent writes to the database from the old system during the upgrade.
2. Run `dbsetup` with the `--dburl`, `--dbuser`, and `--dbpass` options at a minimum. For example:

```
> cd /opt/Axway-7.6.2/analytics/posix/bin

> ./dbsetup --dburl=jdbc:mysql://127.0.0.1:3306/Reporter --dbuser=root --
dbpass=secret
```

Do not specify `--reinstall` as you are not recreating the database. For more information on `dbsetup` command-line options, see the section on configuring the database for API Gateway Analytics in the *API Gateway Installation Guide*.

The following is an example of the output when upgrading a 7.3.x database to version 7.6.2:

```
Current schema version: 001-topology
Latest schema version: 002-leaf
About to upgrade schema. Please note that this operation may take some time for very
large databases
Schema successfully upgraded to: 002-leaf
```

Rollback strategy

If you have created a new database as part of a rollback strategy, the database URL must contain the name of this new database (for example, `--dburl=jdbc:mysql://127.0.0.1:3306/ReporterNew`).

Available database upgrades

The following upgrades are currently available:

Upgrade Name	Description	Compatibility
`000-initial`	Schema used from version 6.3 up to but not including version 7.0.	Must be upgraded to 7.6.2.
`001-topology`	Schema used from version 7.0 up to but not including 7.4.	Must be upgraded to 7.6.2.
`002-leaf`	Schema used from version 7.4.	Compatible with 7.6.2.

Step 5 - Run upgradeconfig to migrate API Gateway Analytics Entity Store customizations

If you have made customizations to the API Gateway Analytics Entity Store in your old installation (for example, if you have added SSL listeners, setup LDAP authentication, or added new users to the user store), you can migrate these changes to your new installation using the utility `upgradeconfig`, rather than setting them up again in the new installation.

Note If you have not made any changes then you are using a default API Gateway Analytics configuration, and you do not need to run `upgradeconfig`.

The `upgradeconfig` utility is located in:

Windows

```
INSTALL_DIR\analytics\Win32\bin
```

Linux

```
INSTALL_DIR/analytics/posix/bin
```

Note You must not use `upgradeconfig` from the API Gateway installation directory; you must use the version in the API Gateway Analytics directory to upgrade API Gateway Analytics configuration.

To run `upgradeconfig`, perform the following steps:

1. Back up your factory configuration by copying the `conf/fed` directory in the new installation (for example, `/opt/Axway-7.6.2/analytics/conf/fed`) to a backup directory.
2. Run `upgradeconfig` with the `-u` and `-o` options. For example:

```
> cd /opt/Axway-7.6.2/analytics/posix/bin

> ./upgradeconfig -u federated:file:////opt/Axway-
7.3.1/analytics/conf/fed/configs.xml
-o /opt/Axway-7.6.2/analytics/conf/fed
```

upgradeconfig options

The following table describes the `upgradeconfig` options:

Option	Description
`-u`	The URL of configuration to upgrade (for example, `federated:file:///C:/Axway730/analytics/conf/fed/configs.xml`).
`-o`	The output directory which will contain the upgraded configuration. Typically this is the `fed` folder of the new API Gateway Analytics installation (for example, `C:\Axway-7.6.2\analytics\conf\fed`).

Use Policy Studio to change API Gateway Analytics configuration

You cannot use Policy Studio instead of `upgradeconfig` to migrate customizations. However, you can use Policy Studio to view the upgraded configuration:

1. In Policy Studio, select **New Project**. Enter a name for the project and make a note of the project location, then select **From existing configuration**.
2. Browse to the `conf/fed` directory in your new API Gateway Analytics installation (for example, `/opt/Axway-7.6.2/analytics/conf/fed`). The API Gateway Analytics configuration is displayed in Policy Studio.
3. Make any required changes to the configuration and save them (**File > Save**).
4. Exit Policy Studio and copy the contents of the project directory (the location you noted earlier) to the `conf/fed` directory of the new API Gateway Analytics installation.

Tip You can also use Policy Studio to view or change any API Gateway Analytics configuration (for example, a 7.6.2 factory configuration).

Step 6 – Run configureserver to configure your new API Gateway Analytics Entity Store

You can use the `configureserver` utility to configure the database connection for API Gateway Analytics (for example, you can specify the database JDBC connection string, the database user, and the database password).

The `configureserver` utility is located in:

Windows

```
INSTALL_DIR\analytics\Win32\bin
```

Linux

```
INSTALL_DIR/analytics/posix/bin
```

To run `configureserver`, enter the following commands:

```
> cd /opt/Axway-7.6.2/analytics/posix/bin
> ./configureserver
```

You are prompted for various settings. Enter a new value for any setting that you wish to change, or press **Enter** to use the default setting.

For more information on `configureserver`, see the section on configuring API Gateway Analytics in the *API Gateway Installation Guide*.

Rollback strategy

If you have created a new database as part of a rollback strategy, the database URL must contain the name of the new database (for example, `jdbc:mysql://127.0.0.1:3306/ReporterNew`).

Step 7 - Migrate custom reports

If you have created your own custom reports in your old API Gateway Analytics installation, you can migrate them to the new installation by copying the files.

Copy all `.json` files from the directory `conf/emc/analytics/reports` in the old installation, for example:

```
/opt/Axway-7.4.1/analytics/conf/emc/analytics/reports
```

Copy the files to the same location in the new installation, for example:

```
/opt/Axway-7.6.2/analytics/conf/emc/analytics/reports
```

Modify API Services report for API Gateway Analytics 7.3.x and earlier

If you are upgrading from API Gateway Analytics version 7.3.x or earlier, you might need to update any API Services custom reports in the new installation.

A report is an API Services report if the report was created with the type `API Service`. In this case, the JSON file will contain:

```
"type":"Service"
```

If the `groupBy` field contains the values `[ "CLIENTNAME", "SERVICENAME" ]`, you must add two extra fields `additionalFields` and `hiddenFields`. For example:

```
"groupBy" : [ "CLIENTNAME", "SERVICENAME" ],
"additionalFields" : [ "DISPLAYNAME" ],
"hiddenFields" : [ "CLIENTNAME" ],
```

Migrate other files

If you have made customizations to the `envSettings.props` file in your old API Gateway Analytics installation, you can migrate them to the new installation by copying the files.

Copy the `envSettings.props` file from the `conf` directory in the old installation, for example:

```
/opt/Axway-7.4.1/analytics/conf
```

Copy the file to the same location in the new installation, for example:

```
/opt/Axway-7.6.2/analytics/conf
```

Step 8 – Stop the old version of API Gateway Analytics and start the new version

If you have installed API Gateway Analytics7.6.2 on the same machine as the old installation, you must stop the old version before starting the new version. You can use the `analytics` script to start and stop API Gateway Analytics.

The `analytics` script is located in:

Windows

```
INSTALL_DIR\analytics\Win32\bin
```

Linux

```
INSTALL_DIR/analytics/posix/bin
```

For example, to stop API Gateway Analytics in the old installation, enter the following commands:

```
> cd /opt/Axway-7.4.1/analytics/posix/bin
> ./analytics -k
```

For example, to start API Gateway Analytics in the 7.6.2 installation, enter the following commands:

```
> cd /opt/Axway-7.6.2/analytics/posix/bin
> ./analytics -d
```

You can now launch the API Gateway Analytics web application. Enter the following address in your browser:

```
http://HOST:8040/
```

`HOST` refers to the host name or IP address of the machine on which API Gateway Analytics is running (for example, `http://localhost:8040/`).

Step 9 – Enable metrics using managedomain

If you have not already done so, you must use the `managedomain` tool to enable the Node Manager to process event logs from your API Gateway host, and to write metrics data to the metrics database.

All API Gateway instances running on the host node generate transaction event log files. These files are all written to the same folder, and are collectively processed and aggregated by the Node Manager on the host, and then written to the metrics database. The metrics database provides the data for the graphical charts in the **Monitoring** view in API Manager and API Gateway Analytics.

Note If you are upgrading API Gateway Analytics from 7.4.0, 7.4.1, or 7.4.2 and you were already using metrics, you do not need to perform this step. If you are upgrading from a version earlier than 7.4.0, you must perform this step.
If you have created a new database as part of a rollback strategy (for example, `ReporterNew`), you must perform this step to specify the database URL of the newly created database to `managedomain`.

In versions earlier than 7.4.0, the API Gateway instance was responsible for writing metrics data to the database. In version 7.4.0 and later, the Node Manager is responsible for writing metrics data to the database.

Before you enable metrics in the Node Manager, you must ensure that the new API Gateway Analytics service is running. Additionally, if you have a multi-node domain with multiple Node Managers, you must ensure that metrics is enabled for each Node Manager.

To enable metrics in the new installation, you can use the `managedomain` utility.

The `managedomain` utility is located in:

Windows

```
INSTALL_DIR\apigateway\Win32\bin
```

Linux

```
INSTALL_DIR/apigateway/posix/bin
```

The following example shows how to modify the host to enable metrics and specify the database settings:

```
./managedomain --edit_host  --metrics_enabled y --metrics_dburl
jdbc:mysql://127.0.0.1:3306/Reporter
```

```
--metrics_dbuser root --metrics_dbpass secretpass
```

The following is an example of the output:

```
There is only one Node Manager in this domain so it must remain as an Admin Node
Manager
Testing Database connectivity for : jdbc:mysql://127.0.0.1:3306/Reporter, user :
root
Metrics database connectivity succeeded
Metrics generation enabled. All other specified metrics settings updated.
Metrics settings updated successfully. Please reboot Node Manager on completion of
this program.
Completed successfully.
```

After you have enabled metrics using the `managedomain` utility, you must restart the Node Manager service. When the Node Manager service has restarted, the upgrade of API Gateway Analytics is complete.

managedomain options

The following table describes the `managedomain` options used:

Option	Description
`--edit_host`	Modify the host settings.
`--metrics_enabled`	Flag to enable or disable metrics. Set to `y` to enable metrics, or set to `n` to disable.
`--metrics_dburl`	The JDBC connection string for the API Gateway Analytics database.
`--metrics_dbuser`	The database user name.
`--metrics_dbpass`	The password associated with the database user name.

For more information on using the `managedomain` utility to enable metrics, see the monitoring and reporting section of the *API Gateway Administrator Guide*.

Rollback strategy

If you have created a new database as part of a rollback strategy, the database URL must contain the name of the new database (for example, `jdbc:mysql://127.0.0.1:3306/ReporterNew`).

Perform a rollback

If you have implemented the rollback strategy described in the preceding sections, follow these steps to revert back to the old version of API Gateway Analytics:

1. Stop the API Gateway Analytics service, the API Gateways, and the Node Managers in the new installation.
2. Restart the API Gateway Analytics service, the API Gateways, and the Node Managers in the old installation.

Further information

For more details on using API Gateway Analytics, see the *API Gateway Analytics User Guide*.

Upgrade your metrics database for API Manager 6

If you are using a metrics database with an earlier version of API Gateway for monitoring in API Manager or third partly tools, you must follow the steps in this section to upgrade the database. Data in your old metrics database will be preserved and the upgraded database will be fully compatible with version 7.6.2.

Note For details on upgrading a metrics database for use with API Gateway Analytics, see *Upgrade API Gateway Analytics* on page 70 instead.

For frequently asked questions about upgrading your metrics database, see *API Gateway Analytics and metrics database upgrades* on page 118.

Summary of steps

The following summarizes the steps to upgrade your metrics database, depending on what version you are upgrading from.

1. Back up your old metrics database.
2. If your version of API Gateway is earlier than 7.4.0, run `dbsetup` to upgrade your database. Otherwise, skip to the next step.
3. If you are not upgrading from 7.4.0, 7.4.1, or 7.4.2 with metrics already enabled, run `managedomain` to enable metrics for Node Managers, if this is not already done. You must also run `managedomain` to update the database URL if you created a new database (for example, as part of a rollback strategy).

Rollback strategy

If you want to be able to revert back to your old version of API Gateway, the best approach is to create a new metrics database for version 7.6.2. The old versions can then be relaunched without changes. Where appropriate, this topic details additional tasks that you need to perform to implement this rollback strategy.

Step 1 - Back up the metrics database in your old installation

You must backup the metrics database being used in the old installation before upgrading. For more details on backing up your database, see your database user documentation.

Rollback strategy

To rollback, you must perform the relevant SQL Restore operation to overwrite the newly created blank database (for example, `MetricsNew`) with the data and schema from the old metrics database.

Step 2 - Run dbsetup to upgrade the metrics database (with versions earlier than 7.4.0)

Note If you are upgrading from API Gateway version 7.4.0 or later, you do not need to run `dbsetup` to upgrade the metrics database as it is already compatible with version 7.6.2. You can skip to the next step.

You can use the utility `dbsetup` to upgrade the metrics database schema and data. This enables you to retain your old database data, but upgrade your database so that it is compatible with API Gateway7.6.2.The `dbsetup` utility always checks the existing version, and modifies the database only if an update is required.

The `dbsetup` utility is located in:

```
INSTALL_DIR/apigateway/posix/bin
```

To upgrade the database:

1. Stop any API Gateway processes or services in the old installation. This is necessary to prevent writes to the database from the old system during the upgrade.
2. Run `dbsetup` with the `--dburl`, `--dbuser`, and `--dbpass` options at a minimum. For example:

```
> cd /opt/Axway-7.6.2/apigateway/posix/bin

> ./dbsetup --dburl=jdbc:mysql://127.0.0.1:3306/Metrics --dbuser=root --
dbpass=secret
```

Note Do not specify `--reinstall` because you are not recreating the database. For more information on `dbsetup` command-line options, see "Configure the metrics database" in the *API Gateway Installation Guide*.

The following is an example of the output when upgrading a 7.3.x database to version 7.6.2:

```
Current schema version: 001-topology
Latest schema version: 002-leaf
About to upgrade schema. Please note that this operation may take some time for very
large databases
Schema successfully upgraded to: 002-leaf
```

Rollback strategy

If you have created a new database as part of a rollback strategy, the database URL must contain the name of this new database (for example, `--dburl=jdbc:mysql://127.0.0.1:3306/MetricsNew`).

Available database upgrades

The following upgrades are currently available:

Upgrade Name	Description	Compatibility
`000-initial`	Schema used from version 6.3 up to but not including version 7.0.	Must be upgraded to 7.6.2.
`001-topology`	Schema used from version 7.0 up to but not including 7.4.	Must be upgraded to 7.6.2.
`002-leaf`	Schema used from version 7.4.	Compatible with 7.6.2.

Step 3 - Enable metrics using managedomain

Note If you are upgrading API Gateway from 7.4.0, 7.4.1, or 7.4.2, and you were already using metrics, you do not need to perform this step. If you are upgrading from a version earlier than 7.4.0, you must perform this step.
If you have created a new database as part of a rollback strategy (for example, `MetricsNew`), you must perform this step to specify the database URL of the newly created database to `managedomain`.

If you have not already done so, you must use the `managedomain` tool to enable the Node Manager to process event logs from your API Gateway host, and to write metrics data to the metrics database.

All API Gateway instances running on the host node generate transaction event log files. These files are all written to the same folder, and are collectively processed and aggregated by the Node Manager on the host, and then written to the metrics database. The metrics database provides the data for the graphical charts in the **Monitoring** view in API Manager, or for third-party tools such as Splunk. For more details on the transaction event log , see the *API Gateway Administrator Guide*.

Note In versions earlier than 7.4.0, the API Gateway instance was responsible for writing metrics data to the database. In version 7.4.0 and later, the Node Manager is responsible for writing metrics data to the database.

To enable metrics in the new installation:

1. You must ensure that the new API Gateway service is running. Additionally, if you have a multi-node domain with multiple Node Managers, you must ensure that metrics is enabled for each Node Manager.
2. Enable metrics using the `managedomain` utility in the following directory:

```
INSTALL_DIR/apigateway/posix/bin
```

The following example shows how to modify the host to enable metrics and specify the database settings:

```
./managedomain --edit_host  --metrics_enabled y --metrics_dburl
jdbc:mysql://127.0.0.1:3306/Metrics
--metrics_dbuser root --metrics_dbpass secretpass
```

The following is an example of the output:

```
There is only one Node Manager in this domain so it must remain as an Admin Node
Manager
Testing Database connectivity for : jdbc:mysql://127.0.0.1:3306/Metrics, user : root
Metrics database connectivity succeeded
Metrics generation enabled. All other specified metrics settings updated.
Metrics settings updated successfully. Please reboot Node Manager on completion of
this program.
Completed successfully.
```

3. Restart the Node Manager service. When the Node Manager service has restarted, the upgrade of metrics database is complete.

managedomain options

The following table describes the `managedomain` options used:

Option	Description
`--edit_host`	Modify the host settings.

Option	Description
`--metrics_enabled`	Flag to enable or disable metrics. Set to `y` to enable metrics, or set to `n` to disable.
`--metrics_dburl`	The JDBC connection string for the metrics database.
`--metrics_dbuser`	The database user name.
`--metrics_dbpass`	The password associated with the database user name.

For more information on using the `managedomain` utility to enable metrics, see the monitoring and metrics section in the *API Gateway Administrator Guide*.

Rollback strategy

If you have created a new database as part of a rollback strategy, the database URL must contain the name of the new database (for example, `jdbc:mysql://127.0.0.1:3306/MetricsNew`).

Perform a rollback

If you have implemented the rollback strategy described in the preceding sections, follow these steps to revert back to the old version of API Gateway:

1. Stop the API Gateways and the Node Managers in the new installation.
2. Restart the API Gateways and the Node Managers in the old installation.

Further information

For more details on API Manager monitoring metrics, see the *API Manager User Guide*.

Resolve upgrade issues in Policy Studio 7

Policy Studio provides graphical features to help you detect and analyze upgrade issues. You can use the **Tools > Upgrade Log Analysis** option to analyze an `upgrade.log` file. You can also use the tool to analyze upgrade issues when creating a new project from existing configuration, or when importing a configuration fragment. For more information, see "Upgrade log analysis" in the *API Gateway Policy Developer Guide*.

You can also use Policy Studio to resolve configuration issues. If the upgrade process identifies issues with the configuration in your old installation, you can use Policy Studio to resolve the issues. There are two ways of doing this:

1. Resolve the issues in the upgraded configuration during the upgrade, and deploy the updated configuration to the new installation when the upgrade completes. See *Resolve issues during upgrade* on page 86.
2. Resolve the issues in the upgraded configuration after the upgrade completes, and deploy the changes. See *Resolve issues after upgrade* on page 86.

Resolve issues during upgrade

During the upgrade, if the `upgrade` command identifies configuration issues that must be resolved in Policy Studio, open the problematic configuration in Policy Studio. For example, in Policy Studio version 7.6.2, create a new project using **From .fed file** or **From .pol and .env files**, and select the correct files under the following directory:

```
/opt/Axway-
7.6.2/apigateway/upgrade/bin/out/upgrade/esgroups/groups/group-2/
```

Modify the problematic configuration in Policy Studio, and save the project for deployment after `sysupgrade` completes.

Tip If Policy Studio is not installed on the node that has the problematic configuration, copy the directory to a machine where Policy Studio is installed, and edit it there.

Resolve issues after upgrade

Alternatively, you can choose to complete the upgrade first and then resolve the configuration issues in Policy Studio. In this case, load the configuration from the running API Gateway after the upgrade completes. For example:

1. In Policy Studio version 7.6.2, click **New Project from an API Gateway instance**.
2. Select the API Gateway group that has the problematic configuration.
3. Make changes to the configuration in Policy Studio, and deploy them.
4. Repeat this process for each API Gateway group in the topology that the upgrade step identified as problematic.

Note You must take care to deploy the correct API Gateway project to the correct API Gateway group.

sysupgrade command reference 8

To perform an upgrade, API Gateway provides a script called `sysupgrade`. This script is located in the following directory:

```
NEW_INSTALL_DIR/apigateway/upgrade/bin
```

The following `sysupgrade` commands are required:

- *export command* on page 88
- *upgrade command* on page 90
- *apply command* on page 93

The following commands are optional:

- *status command* on page 97
- *clean command* on page 98

Note For a description of all available command options and default settings, run `sysupgrade` with the `--help` option. You can also run each command with the`--help` option (for example, `sysupgrade export --help`).

export command

You must run the `export` command first. This contacts the old API Gateway installation using the old Admin Node Manager, and exports configuration data to the `export` output directory. The old installation Admin Node Manager can be running on the same node on which you are running the `export` command, or it can be running on a remote host.

This command also copies local files from the old installation into the `export` output directory. For this reason, it must run locally on each node in a multi-node system.

export command rules

The following rules apply to the `export` command:

- You must run `export` first, and without errors, before running `upgrade`.
- The old installation of the API Gateway system must be running for the `export` command to succeed.

- In a multi-node system:
  - You must run `export` on all nodes, and before running `apply` on any node.
  - It does not matter which node you choose to run `export` on first.

export command options

The following table summarizes the `export` command options:

Option	Description	Required
`--help`	Display help for the `export` command only.	-
`--old_install_dir`	Old API Gateway installation directory. For example: `/home/axway/Axway-7.4.1/apigateway`	Yes
`--anm_host`	Specify the host name of the Admin Node Manager in the old API Gateway installation you are using to export the data. Specify the topology host name of the first Admin Node Manager host you upgrade, and use the same value on every node.	Only required if multiple Admin Node Managers in the topology being upgraded.
`--force`	Force another export when the `export` command has already run successfully. When you specify `--force`, the output from all commands is cleaned and `export` runs again.	No
`--username`	Specify the Admin Node Manager user name.	Yes (prompted if not specified).
`--password`	Specify the Admin Node Manager password.	Yes (prompted if not specified).
`--tracelevel`	Level of logging output. Set to `DEBUG`, `INFO` or `VERBOSE` to increase the output in the log files. The default level is `INFO`.	No

Sample export commands

On a single-node system, or a multi-node system with one Admin Node Manager, `export` is typically run as follows:

```
./sysupgrade export --old_install_dir /home/axway/Axway-
7.3.1/apigateway
```

On a multi-node system with multiple Admin Node Managers, `export` is typically run as follows:

```
./sysupgrade export --old_install_dir /home/axway/Axway-
7.3.1/apigateway --anm_host NodeA
```

To rerun `export` on a node where it has already run successfully (single-node or multi-node with one Admin Node Manager):

```
./sysupgrade export --old_install_dir /home/axway/Axway-
7.3.1/apigateway --force
```

To run `export`and specify admin credentials on the command line (single-node or multi-node with one Admin Node Manager):

```
./sysupgrade export --old_install_dir /home/axway/Axway-
7.3.1/apigateway --username admin --password my_text
```

export command output

The `export` command outputs are as follows:

Output	Location
`export` log file	`Axway-7.6.2/apigateway/upgrade/bin/out/logs/export.log`
`export` output directory	`Axway-7.6.2/apigateway/upgrade/bin/out/export`

For more details on what happens when you run the `export` command, see *Frequently asked questions* on page 110.

upgrade command

You must run the `upgrade` command after `export`, and without errors, before running `apply`.

`upgrade` is an offline command that first performs validation on the exported data. After validation, it upgrades the exported data into a format suitable for import into an API Gateway version 7.6.2 installation. The upgraded data is written to the `upgrade` output directory (in a different location to the exported data).

If any warnings or errors occur during `upgrade`, you are prompted to examine the upgrade log file.

You can resolve warnings or errors with the API Gateway configuration using Policy Studio. For more details, see *Resolve upgrade issues in Policy Studio* on page 86. For examples of other warnings and errors, see *sysupgrade error reference* on page 100.

If any errors occur, you cannot run the `apply` command.

upgrade command rules

The following rules apply to the `upgrade` command:

- You must first run `export` successfully on the local node before you can run `upgrade`.
- You must run `upgrade` on the local node without generating errors, before you can run `apply`.
- You do not need to have the old installation of API Gateway running to run `upgrade`, but you can leave the old API Gateway installation running to minimize downtime.
- In a multi-node system:
  - You must run `upgrade` on all nodes.
  - It does not matter on which node you run `upgrade` first.

upgrade command options

The following table summarizes the `upgrade` command options:

Option	Description	Required
`--help`	Display help for the `upgrade` command only.	-
`--force`	Force another upgrade when the `upgrade` command has already run successfully. When you specify `--force`, the output from `upgrade` and `apply` (if you have already run it) are cleaned, and `upgrade` runs again.	No
`--tracelevel`	Level of logging output. Set to `DEBUG`, `INFO`, or `VERBOSE` to increase the output in the log files. The default level is `INFO`.	No
`--cass_host`	Specify the address of the Apache Cassandra node that will receive data during the `apply` step.	No. Defaults to localhost.

Option	Description	Required
`--cass_port`	Specify the Apache Cassandra client port that will receive data during the `apply` step.	No. Defaults to the Cassandra client default port.
`--cass_username`	Specify the Apache Cassandra user name.	No. Defaults to the Cassandra default user name.
`--cass_password`	Specify the Apache Cassandra password.	No. Defaults to the Cassandra default password.
`--no_cassandra`	If you specify this option, `sysupgrade` checks all group configurations for any Apache Cassandra-backed KPS collections. If any are found, you must update the old installation to remove them or move them to a non-Cassandra data source. For more information, see *Delete or update Apache Cassandra-backed KPS collections in the old installation* on page 25. You must then rerun `sysupgradeexport` and `sysupgrade upgrade --no_cassandra`. If you do not specify this option, an Apache Cassandra host is added to all group configurations. The host details come from the options `--cass_host`, `--cass_port`, `--cass_username`, and `--cass_password`. If not specified, the host and port default to `localhost:9042`.	No.

Sample upgrade commands

On a single-node system, or a multi-node system with one or multiple Admin Node Managers, the `upgrade` command is typically run as follows:

```
./sysupgrade upgrade
```

You can also run this command for a standalone, single-node, Apache Cassandra setup (for example, in a development environment).

To upgrade on a node where the Apache Cassandra server is remote or listening on an address that is not `localhost`:

```
./sysupgrade upgrade --cass_host 192.0.2.0
```

To upgrade if you are not using Apache Cassandra:

```
./sysupgrade upgrade --no_cassandra
```

To rerun `upgrade` on a node where it has already run successfully:

```
./sysupgrade upgrade --force
```

upgrade command output

The `upgrade` command outputs are as follows:

Output	Location
`upgrade` log file	`Axway-7.6.2/apigateway/upgrade/bin/out/logs/upgrade.log`
`upgrade` output directory	`Axway-7.6.2/apigateway/upgrade/bin/out/upgrade`

For more details on what happens when you run the `upgrade` command, see *Frequently asked questions* on page 110.

apply command

The `apply` command first updates any databases used for OAuth or KPS (if an update is required). It then creates and starts the version 7.6.2 Node Manager on the local machine, followed by any API Gateway instances that run locally. `apply` also imports KPS data into the API Gateway instances if required.

In a single-node system, when `apply` runs successfully, `sysupgrade` is complete.

Note `apply` does not update the metrics database. For more details, see *Upgrade API Gateway Analytics* on page 70 or *Upgrade your metrics database for API Manager* on page 81.

In a multi-node system, you must run `apply` on each node. The first node that you run `apply` on must be an Admin Node Manager. When you run `apply` on all subsequent nodes, it registers new Admin Node Managers, Node Managers, and API Gateway instances using the version 7.6.2 Admin Node Manager running on the first node where the `apply` command ran. The first Admin Node Manager to be upgraded holds the domain CA private key and certificate.

Tip When `apply` runs on a subsequent node, it tries to clean any topology entries relating to itself using the first Admin Node Manager, if any entries exist. This ensures it can always be created successfully, even if `apply` ran previously on that node.

apply command rules

The following rules apply to the `apply` command:

- You must run `upgrade` successfully on the local node before you can run `apply`.
- The `apply` command completes the `sysupgrade` procedure on the local node.
- In a single-node system, you must shut down all processes in the old API Gateway installation before running `apply`.
- In a multi-node system:
  - You must run `apply` on all nodes.
  - The first node that `apply` runs on must be an Admin Node Manager.
  - You must shut down the old API Gateway installation on all nodes before running `apply` on any node.
  - `apply` must complete on the first Admin Node Manager node before you run `apply` on subsequent nodes. You can specify the first Admin Node Manager using `--anm_host`. You must use the same `--anm_host` value on each node. You are only required to specify `--anm_host` in a multi-node system with multiple Admin Node Managers.
  - Before you can run the `apply` command on all subsequent nodes, you must run the version 7.6.2 Admin Node Manager on the node you first ran `apply` on. This is the host specified by `--anm_host`. The `apply` command checks that the Admin Node Manager is running and is version 7.6.2.

apply command options

The following table summarizes the `apply` command options:

Option	Description	Required
`--help`	Display help for the `apply` command only.	-

Option	Description	Required
`--anm_host`	Specify the topology host name of the first Admin Node Manager that you are upgrading. You must use the same value on every node.	Only required if multiple Admin Node Managers in upgraded topology.
`--domain_passphrase`	Specify your API Gateway domain CA private key passphrase. You can also set this after upgrading with the `managedomain --change_domain_passphrase` option. For more details, see the managedomain section of the *API Gateway Administrator Guide*.	Yes (prompted if not specified).
`--sign_alg`	Signing algorithm for topology SSL certificates in the new API Gateway system (for example, `sha1`, `sha256`, `sha384`, or `sha512`).	No. Defaults to `sha256`.
`--subj_alt_name`	Specify subject alternative names for Node Manager topology SSL certificates. You can specify multiple subject alternative names. For example: `--subj_alt_name "DNS.0=node1.example.com" --subj_alt_name "DNS.1=host1.example.com" --subj_alt_name "IP.0=10.4.6.7.3" --subj_alt_name "email.0=user1.example.com" --subj_alt_name "email.1=user1.example.com" --subj_alt_name "email.2=user1.example.com"` For more details, see the OpenSSL documentation: https://www.openssl.org/docs/manmaster/apps/x509v3_config.html.	No. Defaults to values suitable for the local host (for example, `DNS.1 = node1.axway.com`, `IP.1 = 10.4.5.6`).
`--force`	Force another `apply` when the `apply` command has already run successfully. When you specify `--force`, the output from `apply` is cleaned and `apply` runs again.	No
`--username`	Specify the Admin Node Manager user name.	Yes (prompted if not specified).
`--password`	Specify the Admin Node Manager password.	Yes (prompted if not specified).

Option	Description	Required
`--skip_db_upgrade`	Skip the upgrade of Key Property Store (KPS) and OAuth tables in external databases. Only use this option if a database upgrade has already been performed (for example, by a database administrator).	No
`--tracelevel`	Level of logging output. Set to `DEBUG`, `INFO`, or `VERBOSE` to increase the output in the log files. The default level is `INFO`.	No

Sample apply commands

On a single-node system, or a multi-node system with one Admin Node Manager, the `apply` command is typically run as follows:

```
./sysupgrade apply
```

On a multi-node system with multiple Admin Node Managers, the `apply` command is typically run as follows:

```
./sysupgrade apply --anm_host NodeA
```

To rerun `apply` on a node where it has already run successfully (single-node or multi-node with one Admin Node Manager):

```
./sysupgrade apply --force
```

To run `apply`and specify admin credentials on the command line (single-node or multi-node with one Admin Node Manager):

```
./sysupgrade apply --username admin --password my_text1 --domain_passphrase my_text2
```

To run `apply` and specify a non-default signing algorithm for the topology SSL certificates (single-node or multi-node with one Admin Node Manager):

```
./sysupgrade apply --sign_alg sha512
```

To run `apply` and specify non-default subject alternative names and a non-default CN for the Node Manager topology SSL certificate (single-node or multi-node with one Admin Node Manager):

```
./sysupgrade apply --subj_alt_name "DNS.0=node1.axway.com" --subj_alt_name
"DNS.1=host1.axway.com"
--subj_alt_name "IP.0=10.4.6.7.3" --subj_alt_name "email.0=user1.axway.com" --subj_
alt_name "email.1=user1.axway.com"
--subj_alt_name "email.2=user1.axway.com" --domain_name MyTestDomain
```

apply command output

The `apply` command outputs are as follows:

Output	Location
`apply` log file	`Axway-7.6.2/apigateway/upgrade/bin/out/logs/apply.log`
`apply` output directory	`Axway-7.6.2/apigateway/upgrade/bin/out/apply`

For more details on what happens when you run the `apply` command, see *Frequently asked questions* on page 110.

status command

The `status` command gets the local node status for `sysupgrade`, For example, which commands (`export`, `upgrade`, and `apply`) have run, and which command should be run next.

Note Before commencing any step, it is recommended that you run the `status` command. This is particularly important in a multi-node upgrade where it is easy to lose track of what step or machine you are on.

For example, the output of the `status` command for each stage of the upgrade process is as follows:

```
./sysupgrade status
System Upgrade Status
---------------------
Current status: not started
Next step: sysupgrade export

./sysupgrade export --old_install_dir=/path/to/old/apigateway
./sysupgrade status
System Upgrade Status
---------------------
```

```
Current status: export complete
Next step: sysupgrade upgrade

./sysupgrade upgrade
./sysupgrade status
System Upgrade Status
---------------------
Current status: export and upgrade complete
Next step: sysupgrade apply


./sysupgrade apply
./sysupgrade status
System Upgrade Status
---------------------
Current status: system upgrade complete
```

clean command

The `clean` command enables you to restart the `sysupgrade` process. The command resets the new API Gateway 7.6.2 installation to factory settings. Any exported data, upgraded data, or recreated topology is moved to a backup directory. You can then rerun all the commands without using the `--force` option.

The `clean` command does the following:

1. Moves the `sysupgrade` output from the `Axway-7.6.2/apigateway/upgrade/bin/out` directory to a backup directory.
2. Removes the `Axway-7.6.2/apigateway/groups` directory from the new API Gateway 7.6.2 installation.
3. Moves Node Manager directories to a backup directory:
    - `apigateway/conf` is moved to `apigateway/upgrade_backup/old-nm_conf-<date>_<time>`
    - `apigateway/system/conf` is moved to `apigateway/upgrade_backup/old-nm_systemconf-<date>_<time>`
    - `apigateway/ext/lib` is backed up to `apigateway/upgrade_backup/old-nm_extlib-<date>_<time>`
4. Restores the original Node Manager directories:
    - `apigateway/conf` is restored from `apigateway/upgrade_backup/original_nm_conf` (but existing `apigateway/conf/licenses` is retained)

- `apigateway/system/conf` is restored from `apigateway/upgrade_backup/original_nm_systemconf`
- `apigateway/ext/lib` is restored from `apigateway/upgrade_backup/original_nm_extlib`

When you run `clean` for the first time:

- Original `apigateway/conf` is backed up to: `apigateway/upgrade_backup/original_nm_systemconf`
- Original `apigateway/system/conf` is backed up to: `apigateway/upgrade_backup/original_nm_systemconf`
- Original `apigateway/ext/lib` is backed up to: `apigateway/upgrade_backup/original_nm_extlib`

clean command rules

The following rules apply to the `clean` command:

- You do not have to run any other step before running `clean`.
- In a multi-node upgrade, you must run `clean` on all nodes to clean the entire system.

clean command options

The following table summarizes the `clean` command options:

Option	Description	Required
`--help`	Display help for the `clean` command only.	-
`--tracelevel`	Level of logging output. Set to `DEBUG`, `INFO`, or `VERBOSE` to increase the output in the log files. The default level is `INFO`.	No

Sample clean command

On a single-node system, or a multi-node system with one or multiple Admin Node Managers, the `clean` command is typically run as follows:

```
./sysupgrade clean
```

For more details on what happens when you run the `clean` command, see *Frequently asked questions* on page 110.

sysupgrade error reference 9

This section describes example warnings and errors generated by the `sysupgrade` script and provides recommended solutions.

Export command errors and warnings

This section describes example errors and warnings that the `sysupgrade export` command generates when exporting the data from the old API Gateway installation.

Errors and warnings are logged to the following file:

```
Axway-7.6.2/apigateway/export/bin/out/logs/export.log
```

Check for customizations to jvm.xml and service.xml

Every API Gateway system stores JVM settings in `VDISTDIR/system/conf/jvm.xml`. Configuration settings for individual API Gateways are stored in `VINSTDIR/conf/service.xml` (for example, `apigateway/groups/group-2/instance-1/conf/service.xml`). If you have modified these files in the old installation, the changes must be manually copied to the corresponding files on the new installation. The `export` step logs an informational message to inform you of this.

Each individual API Gateway instance can also have its own `jvm.xml` file at `VINSTDIR/conf/jvm.xml`. This file is not present by default, but can be created by users. Instance-specific `jvm.xml` files are migrated to the new installation as part of the upgrade process. The `export` step logs a warning to inform you to review these files and verify if they are still needed on the new installation. If not, remove them from the `export` directory before proceeding with the upgrade.

The following is an example of output from the `export` log file:

```
[System Config] : OK
[INFORMATIONAL] messages from [System Config] :
Not migrating the following instance-specific service.xml files:
/opt/Axway-7.4.1/apigateway/groups/group-2/instance-1/conf/service.xml
If you have modified any of these files, you should copy your changes to the new
system after upgrade is complete.
```

```
Not migrating apigateway/system/conf/jvm.xml. This is specific to the old Gateway
version - a new jvm.xml
will be present on the new system. If you have customized this file, you should
manually copy your modifications to the new system.
[WARNING] messages from [System Config] :
WARN: Migrating the following instance-specific jvm.xml files from the old system:
WARN:     /opt/Axway-7.4.1/apigateway/groups/group-2/instance-1/conf/jvm.xml
WARN:     These jvm.xml files are not part of the standard Gateway installation, and
must have been added for
customization. Please review these files and verify that they are still needed on
the new system. If not, they
should be removed from '/home/neil/dev/apigateway/upgrade/bin/out/export/sysconfig'
before proceeding with upgrade.
```

Failure to export data from pre-7.2 installations

Data export can sometimes fail when attempting to upgrade pre-7.2 installations. The `export` step logs the following message describing how to proceed:

```
[KPS] : FAIL
[ERROR] messages from [KPS] :
ERROR: Failed to export KPS data from pre-7.2.0 system. Solution:
ERROR: (1) Create file '/opt/Axway-7.1.0/apiserver/conf/jvm.xml' with the following
contents:
<configurationfragment><classdir
name="$VDISTDIR/upgrade/legacy/7.1.x/"/></configurationfragment>
ERROR: (2) Rerun sysupgrade export.
```

Cassandra configuration

When upgrading from API Gateway versions earlier than 7.5.1, the `export` step checks if there are multiple API Gateways set up with one or more embedded Cassandra servers configured.

If there is only one embedded Cassandra server, the Cassandra configuration summary section simply says:

```
[Cassandra] : OK
```

If there are more than one embedded Cassandra servers and therefore more than one `cassandra.yaml` files present, the `export` step logs a summary similar to the following:

```
[Node Manager Entity Store Configuration] : OK
[Cassandra] : OK
[INFORMATIONAL] messages from [Cassandra] :
In the event of KPS Export errors, confirm that the following files are correctly
```

```
configured :
/opt/Axway-7.4.1/apigateway/groups/group-3/instance-
2/conf/kps/cassandra/cassandra.yaml
/opt/Axway-7.4.1/apigateway/groups/group-3/instance-
3/conf/kps/cassandra/cassandra.yaml
/opt/Axway-7.4.1/apigateway/groups/group-2/instance-
1/conf/kps/cassandra/cassandra.yaml
```

The reason for this message is that there is typically only one API Gateway acting as an embedded Cassandra server. However, you can configure Cassandra to use multiple network cards on the same machine. In this case, there are multiple embedded Cassandra servers (and therefore multiple `cassandra.yaml` files present), and `sysupgrade` logs the above message.

In addition, an incorrect Cassandra configuration might also cause KPS export errors.

The solution in all cases is to ensure that Cassandra is configured correctly in the old installation. For more information, see *Check that embedded Apache Cassandra is configured correctly in the old installation* on page 21.

Inconsistent groups

If your old installation contains any inconsistent groups, the `export` step logs the following message:

```
Group 'TestGroup' (group-2) is inconsistent. Upgrade requires all groups to be
consistent.
```

To resolve this issue, ensure that all API Gateways in the group have the same configuration deployed.

Upgrade command errors and warnings

This section describes example errors and warnings that the `sysupgrade upgrade` command generates when validating the data exported from the old API Gateway installation.

Errors and warnings are logged to the following file:

```
Axway-7.6.2/apigateway/upgrade/bin/out/logs/upgrade.log
```

Admin Node Manager host name is invalid

If the Admin Node Manager has not been configured correctly, the upgrade step logs the following errors:

```
ERROR: Host 'XXX' is not a valid host in the topology. Retry with --anm_host
<hostname>, where hostname is one of: 'NodeA', 'NodeB
ERROR: Host 'nodec' does not have an Admin Node Manager. Retry with a valid --anm_
host <hostname> parameter, where where hostname is one of: 'NodeA', 'NodeB
```

The solution in all cases is to retry the `upgrade` command with a valid `--anm_host` parameter. For more details, see *Which is the first Admin Node Manager?* on page 113

API Manager port changed to HTTPS upgrading from 7.3.1

When upgrading API Gateway version 7.3.1 configuration, the API Manager traffic port (`8065`) is changed from HTTP to HTTPS, and the port is set to use more secure SSL settings. This means that the ciphers are set to `FIPS:!SSLv3:!aNUL`, and the SSLv2/v3 protocols are not allowed. The `upgrade` step logs the following message:

```
The HTTP listener for port PORT.PORTAL.TRAFFIC will be upgraded automatically to a
HTTPS listener.
If you manually reconfigure this listener to HTTP, the Try-It functionality will no
longer work in API Manager.
```

This message is for information only and does not require any action.

Check if API Manager license required

If API Manager is installed on the old API Gateway installation you are upgrading, the `upgrade` step checks the new API Gateway installation for a valid API Manager license. If the license does not exist, the `upgrade` step logs a warning message, but continues the upgrade:

```
WARN: A LicenseException ('feature "apiportal" not licensed') occurred while
checking for a API Manager license in
directory '/opt/Axway-7.6.2/apigateway/conf/licenses'.
Please contact support to acquire a new license. Place the new license in
directory '/opt/Axway-7.6.2/apigateway/conf/licenses'.
```

Check if McAfee license required

If any policy in the old API Gateway installation you are upgrading contains a McAfee Anti-Virus filter, the `upgrade` step checks the new API Gateway installation for a valid McAfee license. If the license does not exist, the `upgrade` step logs a warning message, but continues the upgrade:

```
WARN: A LicenseException ('feature "mcafee" not licensed') occurred while checking
```

```
for a McAfee License
'/opt/Axway-7.6.2/apigateway/'.
Please contact support to acquire a new license. Place the new license in
directory '/opt/Axway-7.6.2/apigateway/'.
```

Check if RBAC permissions file has changed

Each user who logs in to the API Gateway Manager web console has a specific set of roles for Role-Based Access Control (RBAC). These determine what the user can view and what actions they can perform. The roles are defined in the `acl.json` file on the Admin Node Manager in the `apigateway/conf/` directory.

You can manually edit this RBAC permissions file to create new roles or modify existing ones. However, the upgrade does not migrate these changes to the new installation.

The `upgrade` step checks for changes in the RBAC permissions file on the old API Gateway installation. If there are changes, the `upgrade` step logs a warning that the changes need to be manually merged into the `acl.json` file on the new installation:

```
WARN: RBAC permissions file has been modified on old system.
WARN: Differences between original file (/opt/Axway-
7.6.2/apigateway/upgrade/scripts/rbac/factory/acl6.json)
and active file (/opt/Axway-
7.6.2/apigateway/upgrade/bin/out/export/rbac/conf/acl.json)
must be manually merged into apigateway/conf/acl.json on new system.
```

If the RBAC changes are required to allow users to log into the Admin Node Manager or the API Gateway Manager UI, you must do this before you run the `apply` step, or `sysupgrade` will not be able to connect to the new Admin Node Manager and the `apply` step will fail.

Check for corrupt JARs in ext/lib

If a corrupt JAR file is located in the system-wide or instance-specific `ext/lib` directories in the old API Gateway installation, the `upgrade` step logs a warning. The upgrade does not copy the JAR to the new 7.6.2 installation.

Check for old format WSDLs

WSDL files imported to API Gateway versions earlier than 7.4.0 are flagged as having an old resource repository format. This means that they do not take advantage of the improved WSDL resource repository tracking features in later versions of API Gateway. To fix this, remove the web service and reregister the WSDL file. Broken WSDL files are discarded, but the primary WSDL URLs are retained, so you can reimport them in the **Upgrade Log Analysis** view in Policy Studio. For more details, see "Upgrade log analysis" in the *API Gateway Policy Developer Guide*.

Check for valid API Gateway license

API Gateway requires a valid license. If the license does not exist, or is not valid, the `upgrade` step logs one of the following errors:

```
ERROR: A LicenseException ('License not found in /home/axway/Axway-
7.5.0/apigateway/conf/licenses') occurred while checking licenses in
directory '/home/axway/Axway-7.6.2/apigateway/conf/licenses'.
A license may not exist in this directory, or all licenses in this directory may be
invalid.
Please contact support to acquire a new license. Place the new license in
directory '/home/axway/Axway-7.6.2/apigateway/conf/licenses'.
Run sysupgrade again without the --clean option.
```

```
ERROR: A LicenseException ('feature "unrestricted" not licensed') occurred while
checking for a server license in
directory '/home/axway/Axway-7.6.2/apigateway/conf/licenses'.
Please contact support to acquire a new license. Place the new license in
directory '/home/axway/Axway-7.6.2/apigateway/conf/licenses'.
Run sysupgrade again without the --clean option.
```

Note License errors can be generated for a variety of reasons (for example, if it is tampered with, expired, for the wrong product version, or for the wrong host).

You must resolve these errors before you can continue to the `apply` step. If you do not place a valid license in the `conf/licenses` directory before proceeding to the `apply` step, the API Gateway will not start.

SSL certificates for management traffic regenerated

When upgrading to API Gateway 7.6.2 the SSL certificates used for management traffic between Node Managers and API Gateways are always regenerated, so that the old installation and new installation Node Managers cannot communicate with each other.

For example:

```
SSL certificates used between Node Managers and API Gateways for management traffic
will be
regenerated in the upgraded system. Please consult documentation if you wish to
manually reconfigure the upgraded system to use the
SSL certificates from the old version after upgrade.
```

This message is for information only and does not require any action. If you used custom SSL certificates in your old installation, you can manually add the certificates from your old installation to your new installation after upgrade. For more information, contact Axway Support.

Third-party JDBC JARs

If your old installation of API Gateway uses external databases for OAuth and KPS, the `upgrade` step logs the following warning:

```
WARN: OAuth/KPS tables must be upgraded in the following external
databases: [u'jdbc:mysql://127.0.0.1:3306/testdb']. You should copy the required
JDBC drivers to '/opt/Axway-7.5.1/apigateway/upgrade/lib',
so that 'sysupgrade apply' can perform the database upgrades. Alternatively, you can
upgrade OAuth/KPS tables manually using the SQL
scripts at '/opt/Axway-7.5.1/system/conf/sql/upgrades', then run 'sysupgrade apply'
with argument --skip_db_upgrade.
```

Metrics database reconfiguration

If you are using an external database to store API Gateway metrics in your old installation, and your old installation version is earlier than 7.4.0, the `upgrade` step logs the following warning.

```
WARN: Metrics are being written from API Gateway to the following
database: [u'jdbc:mysql://127.0.0.1:3306/testdb']. From 7.4.0 onwards, metrics are
written from the Node Manager, not the API Gateway. You will need
to update the database schema and reconfigure metrics after upgrade.
```

This indicates that you need to reconfigure metrics after the upgrade, because in API Gateway 7.4.0 and later versions, the Node Manager writes metrics instead of the API Gateway. For more information on using `dbsetup` to upgrade the metrics database, see *Upgrade your metrics database for API Manager* on page 81.

ActiveMQ directory

If the old installation of API Gateway uses embedded ActiveMQ, the `upgrade` step might log some of the following warnings.

ActiveMQ directory not set

If the directory for ActiveMQ data is not set in the old installation, the following warning appears:

```
WARN: The ActiveMQ directory is not configured correctly, it is set to '',
sysupgrade will NOT export the data.
```

In this case, `sysupgrade` proceeds with the upgrade. You should fix the ActiveMQ configuration in the old installation.

ActiveMQ directory does not exist

If the directory for ActiveMQ data set in the old installation does not exist, warnings similar to the following might appear.

```
WARN: The absolute ActiveMQ directory '/home/user1/does-not-exist' does not exist,
sysupgrade will NOT export the data.
```

```
WARN: The ActiveMQ directory
'/home/user1/AxwayInstalls/GOLD/7.3.1/apigateway/groups/group-2/instance-1/does-not-
exist'
does not exist, sysupgrade will NOT export the data.
```

In both of these cases, `sysupgrade` proceeds with the upgrade. You should fix the ActiveMQ configuration in the old installation.

ActiveMQ directory not under API Gateway installation directory

If the directory set for ActiveMQ data in the old API Gateway installation is not a directory under the API Gateway installation path, the following warning appears:

```
WARN: The absolute ActiveMQ directory '/home/user1/activemq/data' is NOT a directory
under the old version install
directory '/home/user1/AxwayInstalls/GOLD/7.4.1/apigateway'. The new version system
will use the same directory. When the
new version API Gateway uses the data in this directory it may become unusable by
the old system. You should take a manual
backup of the contents of this directory before proceeding in case you need to run
the old version API Gateway again. Sysupgrade will keep a backup of
the ActiveMQ data in '/home/user1/dev/sysupgrade1_
feature/vordel/install/staging/images/apigateway/upgrade/bin/out/export/activemq/bac
kup'
but this may be removed as a result of a clean command, or export --force.
```

In this case, the upgrade proceeds, and the new API Gateway installation uses the same directory for ActiveMQ data after upgrade.

The new installation might change the data in this directory so that it is no longer usable by the old installation. The `sysupgrade` process makes a backup of the data in the `upgrade/bin/out/export/activemq/backup` directory. However, this backup might be removed if you use the `clean` or `export --force` commands. We recommend that you create an additional backup to another location in case you need to revert back to using the old installation at any stage.

Check for duplicate API Manager data

If API Manager is running in the old installation, exported KPS data is checked for duplicate users called `apiadmin` and duplicate organizations called `Community`. If duplicate data is found, this indicates that KPS is not configured correctly in the old installation.

The `upgrade` step logs the following error message:

```
[ERROR] messages from [API Manager] :
ERROR: Found duplicate 'apiadmin' users in KPS table:
/opt/Axway-7.6.2/apigateway/upgrade/bin/out/upgrade/kps/groups/group-2/instance-
1/conf/kps/backup/sysexport_api_portal_portaluserstoreldap_json
ERROR: Found duplicate 'Community' organizations in KPS table:
/opt/Axway-7.6.2/apigateway/upgrade/bin/out/upgrade/kps/groups/group-2/instance-
1/conf/kps/backup/sysexport_api_portal_portalorganizationstoreldap_json
ERROR: There is a KPS configuration problem on the old system. This must be resolved
before upgrade can proceed. Please contact Axway support for assistance.
```

To resolve this issue, contact Axway Support.

Check for valid FIPS license

If the old API Gateway installation is FIPS-enabled, a FIPS-enabled license is required for the new installation. If the license does not exist, or is not valid, the `upgrade` step logs the following error:

```
ERROR: A LicenseException ('feature "FIPS" not licensed') occurred while checking
for a FIPS license in directory '/home/user1/dev/apigateway/conf/licenses'.
Please contact support to acquire a new license. Place the new license in directory
'/home/user1/dev/apigateway/conf/licenses'.
```

You must resolve this issue before you can continue to the `apply` step.

Check for Apache Cassandra-backed collections

If you run the `sysupgrade upgrade` command with the `--no_cassandra` option, the configuration is checked for Cassandra-backed KPS collections.

If Cassandra-backed collections are found, the `upgrade` step logs the following error:

```
ERROR: Group 'Default Group' contains KPS collections with Cassandra data source:
['My Collection'].
Either run sysupgrade without '--no_cassandra' option, or remove Cassandra
dependency from
source system by deleting collections or moving them to a non-Cassandra data source.
```

You must resolve this error before you can proceed to the `apply` step.

To resolve this error, delete or update the Cassandra-backed KPS collections in your old installation as detailed in *Delete or update Apache Cassandra-backed KPS collections in the old installation* on page 25, and then rerun the `export` and `upgrade` steps.

Alternatively, you can rerun the `upgrade` step without the `--no_cassandra` option. However, your new installation will require an Apache Cassandra database cluster as detailed in *Configure an Apache Cassandra database cluster in the new installation* on page 24.

Apply command errors and warnings

This section describes example errors and warnings that the `sysupgrade apply` command generates when applying the upgraded data to the new API Gateway installation.

Errors and warnings are logged to the following file:

```
Axway-7.6.2/apigateway/apply/bin/out/logs/apply.log
```

Incorrect Admin Node Manager host

If the `--anm_host` specified is not the first Admin Node Manager host and the first Admin Node Manager host is not running, the `apply` step logs the following error:

```
ERROR: Node Manager error: Failed to sign CSR for service. None of the available
Admin Node Managers have the domain private key,
or were able to unlock it with provided passphrase.
```

The solution is to always use the same `--anm_host` value for the first Admin Node Manager. For more details, see *API Gateways missing from topology after upgrade* on page 122.

Frequently asked questions 10

This topic provides answers to several frequently asked questions about upgrade. It includes FAQs for:

- *All upgrades* on page 110
- *Single-node upgrades* on page 111
- *Multi-node upgrades* on page 113
- *API Gateway Analytics and metrics database upgrades* on page 118

All upgrades

The following FAQs are specific to all upgrades.

Why would you rerun export?

You must rerun `export` if:

- The previous attempt to run `export` failed (for example, because the old installation processes were not running).
- You have made a change to the old installation that you also want to apply to the new installation.
- You updated the configuration in the old installation to resolve a warning or error reported when you ran `upgrade`.

Note If you rerun `export`, you must rerun all subsequent steps (`upgrade` and `apply`).

What happens if you change the old API Gateway installation after running export?

If you make any changes to the old API Gateway installation after running `export` and you do not need these changes to be included in the new 7.6.2 installation, you do not need to take any action.

If you do want the changes to be included in the new 7.6.2 installation, you must rerun `export`, possibly on all nodes, depending on the changes made. For example, if you deploy a new configuration to a group of API Gateways, you must rerun `export` on all nodes that run instances in that group.

Note If you rerun `export`, you must rerun all subsequent steps (`upgrade` and `apply`).

If you rerun `export`, and therefore `upgrade` and `apply`, on the first Admin Node Manager, you have cleaned your topology, so you must rerun `apply` on all other nodes.

Why would you rerun upgrade?

You must rerun `upgrade` if:

- You have rerun `export`. For more details, see *Why would you rerun export?* on page 110
- The previous attempt to run `upgrade` failed with errors.

Note When upgrading very large configurations, the default memory settings might not be sufficient for `upgrade` to run successfully (see *Out of memory error when running upgrade* on page 120).

Why would you rerun apply?

You must rerun `apply` if:

- You have rerun `export` or `upgrade`. For more details, see *Why would you rerun export?* on page 110 and *Why would you rerun upgrade?* on page 111
- The previous attempt to run `apply` failed with errors (for example, if you specified the wrong `--anm_host` parameter).

Why would you run clean?

You might choose to run `clean` if:

- Issues occurred during the upgrade process and you want to restart the upgrade.
- You need to restart the upgrade to capture new changes made to the old API Gateway installation.

Note The `export --force` command also triggers a full clean of the `export`, `upgrade`, and `apply` outputs, but in addition it also attempts to run `export`; while the `clean` command only cleans the outputs.

Single-node upgrades

The following FAQs are specific to single-node upgrades.

What happens if you rerun export when you have already run apply?

In a single-node domain, if you have previously run `export` successfully and try to rerun `export`, you are prompted to rerun with the `--force` option. This cleans the `export`, `upgrade`, and `apply` outputs, and then reruns `export` on the node. You must then rerun `upgrade` and `apply` on the node. The `--force` option is not required for `upgrade` or `apply` as the outputs have already been cleaned.

Note Before you rerun `export`, you must ensure that any processes in the new 7.6.2 installation are stopped, and that the processes in the old installation are running.

What happens if you rerun upgrade when you have already run apply?

In a single-node domain, if you have previously run `upgrade` successfully, and try to rerun, you are prompted to rerun with the `--force` option. This cleans the `upgrade` and `apply` command outputs, and then reruns `upgrade` on the node. You must then rerun `apply` on the node. The `--force` option is not required for `apply` as the output has already been cleaned.

Note The `upgrade` command runs offline and does not require any API Gateway processes to be running.

What happens if you rerun apply?

In a single-node domain, if you have previously run `apply` successfully, and try to rerun, you are prompted to rerun `apply` with the `--force` option. This cleans the `apply` command output, and then reruns `apply` on the node.

Note If any API Gateway processes are running in the old or new installation, you are prompted to stop them before `apply --force` can succeed.

What happens if you run clean?

If you run `clean` on a single-node domain, you are back to the start of the `sysupgrade` process. To redo the upgrade, you must:

1. Run `export`.

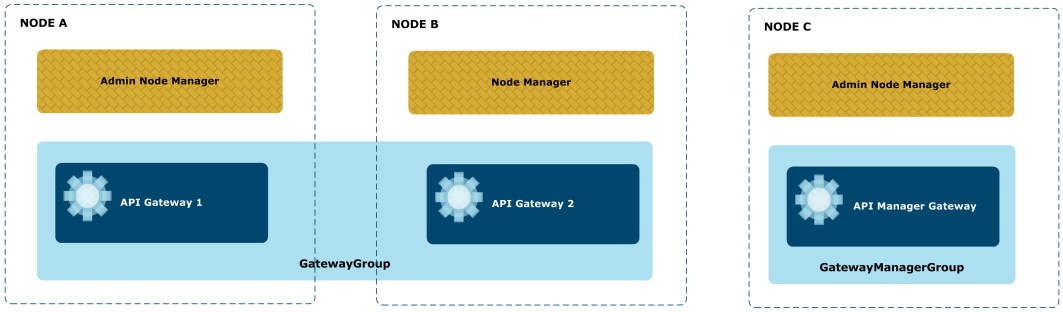
   Note Before you run `export`, you must ensure that all processes in the new 7.6.2 installation are stopped, and that all processes in the old installation are running.

2. Run `upgrade`.
3. Shut down all processes in the old installation.

4. Run `apply`.

Multi-node upgrades

The following FAQs are specific to multi-node upgrades. The example topology referenced in these FAQs is as follows:



Which is the first Admin Node Manager?

In a multi-node domain, the Admin Node Manager that has the domain CA private key file is considered the first Admin Node Manager. To determine if an Admin Node Manager is the first Admin Node Manager, check if the following file exists on the node:

```
apigateway/groups/certs/private/domain.p12
```

In the old installation, the first Admin Node Manager is always the first Node Manager created in the domain. In the new installation, the first Admin Node Manager is always the first node on which you run `apply`, and that node must be an Admin Node Manager in the old API Gateway domain.

Tip The first Admin Node Manager does not have to be on the same node in the old and new API Gateway installations, but this is the simplest approach.

For example, in a three-node domain with Admin Node Managers on NodeA and NodeC (and none on NodeB), where NodeA is the first Admin Node Manager in the old installation, the first Admin Node Manager can be on NodeC in the new installation (if you run `apply` on NodeC first). However, the first Admin Node Manager cannot be on NodeB in the new installation, as NodeB was not an Admin Node Manager in the old installation.

What happens if you rerun export on the first Admin Node Manager and have already run apply?

In a multi-node domain, if you have previously run `export` successfully on the first Admin Node Manager node (for example, NodeA) and try to rerun `export`, you are prompted to rerun with the `--force` option. This cleans the `export`, `upgrade`, and `apply` outputs on NodeA, and then reruns `export` on NodeA. Running `export --force` on the first Admin Node Manager in a multi-node domain also cleans your topology and domain CA private key and certificate.

Note Before you rerun `export`, you must ensure that all processes in the new 7.6.2 installation are stopped *on all nodes*, and that all processes in the old installation are running *on all nodes*.

After running `export --force` on the first Admin Node Manager (for example, NodeA) in a multi-node domain, you must perform the following steps:

1. Rerun `upgrade` on NodeA.

   Note You might also need to rerun `export` and `upgrade` on other nodes. We recommend using the `status` command on all other nodes to ensure that all nodes are at the same stage before proceeding.

2. Shut down all processes in the old installation on all nodes.
3. Rerun `apply` on NodeA.
4. Rerun `apply` on all other nodes, in any order. This step is necessary because running `export --force` on NodeA cleans your topology and domain CA private key and certificate.

What happens if you rerun export on a node that is not the first Admin Node Manager and have already run apply?

In a multi-node domain, if you have previously run `export` successfully on a node that is not the first Admin Node Manager node (for example, NodeC) and try to rerun `export`, you are prompted to rerun with the `--force` option. This cleans the `export`, `upgrade`, and `apply` outputs on NodeC, and then reruns `export` on NodeC. NodeC can be an Admin Node Manager or a Node Manager.

Note Before you rerun `export`, you must ensure that all processes in the new 7.6.2 installation are stopped *on all nodes*, and that all processes in the old installation are running *on all nodes*.

After running `export --force` on a node that is not the first Admin Node Manager node (for example, NodeC) in a multi-node domain, you must perform the following steps:

1. Rerun `upgrade` on NodeC.
2. Shut down all processes in the old installation on all nodes.
3. Start up the processes in the new 7.6.2 installation on other nodes, especially the Admin Node Manager on NodeA, as this must be running before you can proceed to the next step.
4. Rerun `apply` on NodeC. Because you ran `apply` previously on NodeC, this rerun triggers removal of NodeC entries in the topology on the Admin Node Manager on NodeA, before NodeC is registered.

You do not need to rerun commands on other nodes as a result of running `export --force` on NodeC.

What happens if you rerun upgrade on the first Admin Node Manager and have already run apply?

In a multi-node domain, if you have previously run `upgrade` successfully on the first Admin Node Manager node (for example, NodeA) and try to rerun `upgrade`, you are prompted to rerun with the `--force` option. This cleans the `upgrade` and `apply` outputs on NodeA, and then reruns `upgrade` on NodeA. Running `upgrade --force` on the first Admin Node Manager in a multi-node domain also cleans your topology and domain CA private key and certificate.

Note The `upgrade` command runs offline and does not require any API Gateway processes to be running.

After running `upgrade --force` on the first Admin Node Manager (for example, NodeA) in a multi-node domain, you must perform the following steps:

1. Shut down all processes in the old installation on all nodes.

   Note You might also need to rerun `upgrade` on other nodes. We recommend using the `status` command on all other nodes to ensure that all nodes are at the same stage before proceeding.

2. Rerun `apply` on NodeA.
3. Rerun `apply` on all other nodes, in any order. This step is necessary because running `upgrade --force` on NodeA cleans your topology and domain CA private key and certificate.

What happens if you rerun upgrade on a node that is not the first Admin Node Manager and have already run apply?

In a multi-node domain, if you have previously run `upgrade` successfully on a node that is not the first Admin Node Manager node (for example, NodeC) and try to rerun `upgrade`, you are prompted to rerun with the `--force` option. This cleans the `upgrade` and `apply` outputs on NodeC, and then reruns `upgrade` on NodeC. NodeC can be an Admin Node Manager or a Node Manager.

Note The `upgrade` command runs offline and does not require any API Gateway processes to be running.

After running `upgrade --force` on a node that is not the first Admin Node Manager node (for example, NodeC) in a multi-node domain, you must perform the following steps:

1. Shut down all processes in the old installation on all nodes.
2. Shut down all processes in the new 7.6.2 installation on NodeC (for example, any processes started by a previous `apply`).
3. Start up the processes in the new 7.6.2 installation on other nodes, especially the Admin Node Manager on NodeA, as this must be running before you can proceed to the next step.
4. Rerun `apply` on NodeC. Because you ran `apply` previously on NodeC, this rerun triggers removal of NodeC entries in the topology on the Admin Node Manager on NodeA, before NodeC is registered.

You do not need to rerun commands on other nodes as a result of running `upgrade --force` on NodeC.

What happens if you rerun apply on the first Admin Node Manager?

In a multi-node domain, if you have previously run `apply` successfully on the first Admin Node Manager node (for example, NodeA) and try to rerun `apply`, you are prompted to rerun with the `--force` option. This cleans the `apply` output on NodeA and then reruns `apply` on NodeA. Running `apply --force` on the first Admin Node Manager in a multi-node domain also cleans your topology and domain CA private key and certificate.

Note If any API Gateway processes are running in the old or new installation, you are prompted to stop them before `apply --force` can succeed.

After running `apply --force` on the first Admin Node Manager (for example, NodeA) in a multi-node domain, you must perform the following steps:

1. Leave all processes in the new 7.6.2 installation running on NodeA.
2. Shut down all processes in the new 7.6.2 installation on all other nodes (the processes in the

old installation should already be down).

3. Rerun `apply` on all other nodes, in any order.

What happens if you rerun apply on a node that is not the first Admin Node Manager?

In a multi-node domain, if you have previously run `apply` successfully on a node that is not the first Admin Node Manager (for example, NodeC) and try to rerun `apply`, you are prompted to rerun with the `--force` option. This cleans the `apply` outputs on NodeC, and then reruns `apply` on NodeC. NodeC can be an Admin Node Manager or a Node Manager.

Note If any API Gateway processes are running in the old or new installation, you are prompted to stop them before `apply --force` can succeed.

You do not need to rerun commands on other nodes as a result of running `apply --force` on NodeC.

What happens if you run clean on the first Admin Node Manager?

In a multi-node domain, if you run `clean` successfully on the first Admin Node Manager (for example, NodeA), this cleans the `export`, `upgrade`, and `apply` outputs on NodeA. It also cleans your topology and domain CA private key and certificate.

To redo the upgrade on NodeA, you must perform the following:

1. Run `export` on NodeA.

   Note Before you run `export`, you must ensure that all processes in the new 7.6.2 installation are stopped *on all nodes*, and that all processes in the old installation are running *on all nodes*.

2. Run `upgrade` on NodeA.

   Note You might also need to rerun `export` and `upgrade` on other nodes. We recommend using the `status` command on all other nodes to ensure that all nodes are at the same stage before proceeding.

3. Shut down all processes in the old installation on all nodes.
4. Run `apply` on NodeA.
5. Run `apply` on all other nodes, in any order. This step is necessary because running `clean` on NodeA cleans your topology and domain CA private key and certificate.

What happens if you run clean on a node that is not the first Admin Node Manager?

In a multi-node domain, if you run `clean` successfully on a node that is not the first Admin Node Manager (for example, NodeC), this cleans the `export`, `upgrade`, and `apply` outputs on NodeC. Your topology and domain CA private key and certificate still exist on another Admin Node Manager node (for example, NodeA).

To redo the upgrade on the node you have cleaned, you must perform the following steps:

1. Run `export` on NodeC.

   Note Before you run `export`, you must ensure that all processes in the new 7.6.2 installation are stopped *on all nodes*, and that all processes in the old installation are running *on all nodes*.

2. Run `upgrade` on NodeC.
3. Shut down all processes in the old installation on all nodes.
4. Start up the processes in the new 7.6.2 installation on other nodes, especially the Admin Node Manager on NodeA, as this must be running before you can proceed to the next step.
5. Run `apply` on NodeC. If you ran `apply` previously on NodeC, this rerun triggers removal of NodeC entries in the topology on the Admin Node Manager on NodeA, before NodeC is registered.

You do not need to rerun commands on other nodes as a result of running `clean` on NodeC.

API Gateway Analytics and metrics database upgrades

The following FAQs are specific to upgrading API Gateway Analytics and your metrics database.

What should you upgrade first - API Gateway or API Gateway Analytics?

This depends on the version of API Gateway you are upgrading from:

- If you are upgrading from 7.4.0 and later versions, you should upgrade API Gateway Analytics before upgrading API Gateway using the `sysupgrade` command.
- If you are upgrading from versions earlier than 7.4.0, you should upgrade API Gateway Analytics after upgrading API Gateway using the `sysupgrade` command.

For more details, see *Upgrade API Gateway Analytics* on page 70.

Do you need to run managedomain to enable metrics?

If you are upgrading from a pre-7.4.0 API Gateway domain, you must run `managedomain` for each Node Manager in the domain to enable metrics. You must then restart each Node Manager.

If you are upgrading from a 7.4.0 or later API Gateway domain where the Node Managers are configured to write to the database, you do not need to run `managedomain` for each Node Manager. This is because the configuration is migrated when the API Gateways are upgraded.

For details on API Gateway Analytics metrics, see *Step 9 – Enable metrics using managedomain* on page 78.

For details on API Manager metrics, see *Upgrade your metrics database for API Manager* on page 81.

Troubleshoot an upgrade 11

This topic provides advice on troubleshooting the API Gateway upgrade process and the `sysupgrade` commands.

Out of memory error when running upgrade

Problem: When upgrading a large configuration the upgrade command fails with an out of memory error.

Solution: Some configurations are very large and need memory to be increased above the default values.

To increase the memory, perform the following steps:

1. Edit the `-Xmx` setting in the `apigateway/system/conf/jvm.xml` file to specify a value (for example, `2048`) instead of the default.

   The value required depends on the size of your configuration. The following example shows how to change it to `2048`.

   Default `-Xmx` value:

```
<if property="maxHeap">
 <VMArg name="-Xmx${maxHeap}"/>
</if>
```

   `-Xmx` value of `2048`:

```
<if property="maxHeap">
 <VMArg name="-Xmx2048"/>
</if>
```

2. Rerun the `upgrade` command.

Apply was run before export was run on all nodes

Problem: You want to upgrade the remaining nodes in a multi-node domain, but you have already run `export`, `upgrade`, and `apply` on the first Admin Node Manager in the domain.

Solution: In this case, you did not follow the rule that `export` must run on all nodes in a multi-node domain before `apply` is run on any node (see *apply command rules* on page 94).

This means that you have API Gateway processes from the new installation running on the first Admin Node Manager, (for example, NodeA), but have other nodes (for example, NodeB and NodeC) that are still running API Gateway processes from the old installation.

To upgrade NodeB and NodeC, perform the following steps:

1. Shut down the API Gateway processes in the new installation on NodeA and then start up the API Gateway processes in the old installation on NodeA.
2. Ensure that the API Gateway processes from the old installation are still running on NodeB and NodeC.
3. Run `export` on NodeB.
4. Run `upgrade` on NodeB.
5. Run `export` on NodeC.
6. Run `upgrade` on NodeC.
7. Shut down the API Gateway processes in the old installation on all nodes.

   Tip If you need to bring up the processes in the new installation on NodeA more quickly, you can shut down the processes in the old installation on all nodes after `export` is run on NodeB and NodeC.
8. Start the API Gateway processes on the new 7.6.2 installation on NodeA.
9. Run `apply` on NodeB.
10. Run `apply` on NodeC.

Group inconsistency errors

Problem: Group inconsistency error at export, but groups are not inconsistent in the old installation.

Solution: This error occurs if a remote Node Manager is down. Restart the remote Node Manager and rerun the `export` command.

KPS data missing after upgrade

Problem: After running `apply` on all nodes, some KPS data appears to be missing.

Solution: Use the `import-kps` command. This command is available in your API Gateway installation directory (for example, `/opt/Axway-7.6.2/apigateway/upgrade/bin`).

The following table summarizes the `import-kps` command options:

Option	Description	Required
`--help`	Display help for the `import-kps` command only.	-

Option	Description	Required
`--anm_host`	Specify the topology host name of the Admin Node Manager in the old API Gateway installation you are using to export the data. We recommend that you specify the first Admin Node Manager host that you are upgrading, and you must use the same value on every node.	Only required if multiple Admin Node Managers in upgraded topology.
`--username`	Specify the Admin Node Manager user name.	Yes (prompted if not specified).
`--password`	Specify the Admin Node Manager password.	Yes (prompted if not specified).

API Gateways missing from topology after upgrade

Problem: You ran `apply` successfully on all nodes, but API Gateways are missing from your topology.

Solution: If you are upgrading a topology with multiple Admin Node Managers, you must pass the host name of the first Admin Node Manager you upgraded to the `apply` command *on all nodes*.

For example, if you have three nodes as follows:

- **NodeA**: Admin Node Manager and API Gateway1 in Group1
- **NodeB**: Node Manager and API Gateway1A in Group1
- **NodeC**: Admin Node Manager and API Gateway2 in Group2

If you run `apply` incorrectly as follows:

- **NodeA**: `sysupgrade apply --anm_host NodeA`
- **NodeB**: `sysupgrade apply --anm_host NodeA`
- **NodeC**: `sysupgrade apply --anm_host NodeC`

Everything appears to upgrade successfully. However, you now have two separate domains in your new 7.6.2 installation. If you log in to `https://NodeA:8090`, you will see API Gateway1 and API Gateway1A. If you log in to `https://NodeC:8090`, you will see API Gateway2.

To resolve this issue, follow these steps:

1. Stop all API Gateway processes in the new installation on NodeC.
2. Rerun `apply` on NodeC as follows:

```
sysupgrade apply --anm_host NodeA
```

When this command completes, NodeC joins the domain where NodeA is the first Admin Node Manager.

Tip Rerunning apply on NodeC is the easiest solution in this case. Alternatively, you could rerun `apply --anm_host NodeC` on NodeA and on NodeB.

Get help on sysupgrade commands

For a description of all available command options and default settings, run `sysupgrade` with the `--help` option.

You can also run each command with the`--help` option. For example, `sysupgrade export --help`, `sysupgrade upgrade --help`, and so on.