AWS Migration Portfolio Assessment

- User Guide -

AWS Migration Portfolio Assessment (MPA): a user guide

Copyright © 2023 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Features overview ***5***

Getting started ***7***

When to use MPA? 7
Prerequisites 7
Are you a first-time user of MPA? 7

1. Portfolio management ***8***

1.1. Create portfolio 8
1.2. Share portfolio 8
1.3. Edit portfolio properties 8
1.4. Copy a portfolio 9
1.5. Delete portfolios 9
1.6. Navigating a portfolio 9

2. Data entities ***10***

2.1. Servers 10
2.2. Business applications 13
2.2.1. Additional attributes 13
2.3. Databases 13
2.4. Shared storage 14
2.5. Network 15
2.6. Dependencies 15
2.7. Mainframe workloads 15

3. Data management ***16***

3.1. Import from file 16
3.1.1. The import flow 16
3.1.2. Things to consider 17
3.2. Import from ADS 18
3.3. Data collection 19
3.4. Data validation 20
3.5. Data update 21
3.6. Data visualization 21

4. Total cost of ownership ***22***

4.1. AWS recommendation types 22
4.2. Assumptions 23
4.3. Navigating the TCO results 24
4.3.1. TCO summary 24
4.4. EC2 recommendations 26
4.4.1. EC2 sizing 26
4.4.2. Server level EC2 instance preferences 27
4.4.3. EC2 recommendations manual override 28
4.4.4. EC2 cost estimation 28
4.5. Dedicated Host recommendations 29
4.5.1. Dedicated Host sizing 29
4.5.2. Dedicated Host cost estimation 30
4.5.3. Dedicated Host - 3rd party license estimation 30
4.6. Recommendations for EC2 with SQL Server license included 31
4.7. Recommendations for EC2 shared with Oracle bring your own license 32
4.8. RDS recommendations 33

4.8.1. RDS sizing 33
4.8.2. RDS cost estimation 34
4.8.3. Network cost estimation 34
4.9. On-Premises cost estimation 35
4.9.1. Compute 35
4.9.2. Rack infrastructure 35
4.9.3. OS and hypervisor licensing 36
4.9.4. Database 37
4.9.5. Database licensing estimation 37
4.10. Network cost 38
4.11. Assessment comparison 38
4.12. Troubleshooting TCO results 39
4.12.1. EC2 recommendations 39
4.12.2. RDS recommendations 39
4.13. Adjustments 40
4.14. TCO download 41

5. Application cost ***42***
5.1. AWS costs estimation 42
5.1.1. Direct costs 42
5.1.2. Common costs 43
5.2. On-Premises costs estimation 43
5.2.1. Direct costs 43
5.2.2. Common costs 44

6. SQL Server optimization ***46***

7. Mainframe workloads assessments ***47***
7.1. Mainframe workloads annual cost 47
7.2. Migration cost 48
7.3. AWS annual cost 49

8. Migration strategy ***52***
8.1. Servers 52
8.2. Applications 53

9. Migration cost ***54***
9.1. Estimate 54
9.2. Cash flow 55

10. Application prioritization ***57***

11. Dependency groups ***58***

12. Migration wave ***59***

Appendix ***60***
1. PDM steps to issue partner access codes 60

Document history ***62***

Thank you! **64**

Features overview

Portfolio assessment is the process of discovery, identification, classification and grouping of IT infrastructure and applications, to estimate the feasibility, effort, cost, tools and resources needed for migration to Amazon Web Services.

AWS Migration Portfolio Assessment (MPA) is a web application intended to assist in this process.

MPA helps you consolidate the customer infrastructure data in one place, build the business case by providing TCO (total cost of ownership) comparison between On-Premises and AWS, estimate the migration cost, plan the migration through R-type strategies, application prioritization, dependency grouping and wave plans.

On-premises inventory

Everything in MPA happens in a portfolio, which you create and have sole access to. You import into the portfolio application and infrastructure data describing the customer's on-premises estate; you can do this either from files or from ADS (AWS Application Discovery Service).

Data collection

After the initial import you can use the *Data collection* feature to gather additional information about the imported assets. You create asset questionnaires, assign them to customer contacts and request their inputs. Their answers, provided through a separate AWS application, are synced into MPA.

Business case (TCO)

The migration to AWS business case can include cost savings, as well as soft benefits (staff productivity, operational resilience and business agility). MPA helps with the cost savings by estimating the On-Premises costs and providing a *directional* TCO for AWS. The On-Premises costs are based on industry benchmarks from the AWS Cloud Economics team; they account for server and storage cost (including rack and facility cost), database cost (including license estimation), network, and administration cost. The AWS TCO supports compute (either EC2 Shared/Dedicated instance or EC2 Dedicated Host), database cost (RDS and license estimation on Dedicated Host), storage (EBS, EFS, S3/Glacier), network, admin and AWS Support costs. You control which recommendation (EC2, DH, RDS) applies based on the customer needs. You can also tweak 100+ settings, called assumptions, to accurately capture the customer On-Premises details and in-cloud needs. For those costs that MPA can't yet estimate, you can manually enter the amounts as adjustments.

MPA also supports comparing different assessments, allowing you to run what-if analyses.

In addition to the cost savings aspect of the business case, customers often need an estimate of the cost to migrate to AWS. MPA provides that based on the migration strategy of the servers and databases in the portfolio, migration project duration and migration resource mix available to the project. MPA provides default recommendations for the migration strategy of servers based on their OS and hypervisor.

Portfolio analysis

MPA offers some capabilities for analyzing the data in a portfolio. You can visualize the imported servers through customizable charts, assign migration strategy (R-type values) and prioritize applications based on customer preferences.

Migration planning

MPA allows you to create migration waves with configurable capacity and duration. Through dependency grouping, you manage the relations between applications and their resources, creating dependency groups which you then assign to wave plans.

Data access and GDPR

MPA is not part of the AWS Console; using MPA doesn't require an AWS account for you, or your customer. All the portfolios you create and their data are accessible only to you. You can choose to share with others, and delete them at any time. All the data is currently only processed and stored in one AWS Region, us-west-2 (Oregon). While MPA is plugged into the AWS processes for GDPR (General Data Protection Regulation) compliance for handling of DSAR (Data Subject Access Request) and RTF (Right To be Forgotten) requests, should your customer have any data privacy concerns, you can avoid entering any PII (Personal Identifiable Information) data such as server IPs and customer contact details (generic email addresses can be used for data collection). Note that use of MPA is covered by the same AWS side *Privacy Notice* which you can refer to at the bottom of every MPA page.

Note:

- Throughout this document and in MPA itself references to "data" only refer to metadata about the customer workloads and/or target AWS architecture. There is no need and you should never upload in MPA actual data from customer databases for example, only their metadata.
- MPA follows the *Cookie preference* requirements of GDPR. You can toggle your preferences at any time from the footer of the page

. There are few UI capabilities that will not work as they rely on local storage, hence fall under *Functional* cookies.

Getting started

When to use MPA?

MPA can be used during the early stage of the migration discussions, as early as the customer's portfolio data is available. The target architecture on AWS may differ from the On-Premises architecture; use MPA to get a directional sense in the beginning of the process.

Prerequisites

- MPA is intended for cloud migration practitioners, therefore you are either an AWS employee or a partner. At the moment, customers don't have direct access to MPA. Experience in migrations is preferred, but if you are new to the field, you can refer to the *Application portfolio assessment for AWS Cloud migration* (link) to understand the overall *Portfolio Assessment* method. For MPA specific trainings, there are helpful immersion materials available, one of them is here. If you are an AWS partner, it is recommended that you take the *Cloud Economics Accreditation* course which can be found here.

- You have either a customer's CMDB extract or the output of a discovery tool, which can be in files or stored in ADS. For ADS, you need the access details of a user with proper permissions.

- You have customer's authorization to store their On-Premise infrastructure data in MPA. It is recommended to also have the customer's agreement as called out in the *Site terms* at the bottom of every MPA page.

- You are familiar with the AWS services and their pricing options.

- You have a working understanding of the English language. Currently MPA only supports English content. MPA does leverage your browser's locale for language settings, but those only impact how the currency values are presented. (For Chrome, you can manage the default language under chrome://settings/languages).

- Technical requirements: MPA is a web-based application. Works best in Chrome and can face issues if certain browser extensions blocking CORS are enabled. Some firewalls can also cause the import of files to fail.

Are you a first-time user of MPA?

You can access MPA at https://accelerate.amazonaws.com.

AWS partners can sign in with AWS Partner Central, hence use their *APN Portal* (*AWS Partner Network*) credentials. Only *Select*, *Advanced* and *Premier* tiers are allowed access to MPA. For *Registered*, access can be granted (ex: for training purposes) through access codes. Partner Delivery Managers can issue access codes for their partners, see Appendix for details.

After login, launch the Portfolio Assessment option, it will redirect you to the MPA landing page: https://mpa.accelerate.amazonaws.com.

1. Portfolio management

Everything in MPA happens in a *portfolio*, which you create and have sole access to. A portfolio is where the customer data is imported and where all the results are available. You can create as many portfolios as needed and switch freely between them: what you change in one portfolio is not impacting the others. The *All portfolios* page shows the list of those portfolios you have access to, and from here you can take portfolio level actions. You can share the portfolios with others in your team, add tags for easy categorization, copy an existing portfolio into a new one, and delete the portfolio when no longer needed.

1.1. Create portfolio

You create a new portfolio by clicking Get started on the landing page, or navigating to *All portfolios* from pretty much anywhere in MPA. On the *All portfolios* page, click Create portfolio.

Portfolio name and *Account name* are mandatory fields for portfolio creation.
AWS partner users can search for existing customer account or create a new customer account. MPA will only show those existing accounts created by users within the same APN organization as you. It is recommended to create a customer account only once, and use it for all portfolios of that customer within your organization.

If you just want to test MPA, you can create a sample portfolio – it will be pre-populated with sample data.
You can also set tags to help you categorize the portfolios you have in MPA.

1.2. Share portfolio

By default, after creating a portfolio you are the only one that has access to it. You can grant access to other users in your team by selecting the portfolio on the *All portfolios* page, and clicking *Share*. Type the user's email address, select *Full* or *Update* as permission type and click *Add*. MPA sends a notification email to the specified address.

Note:

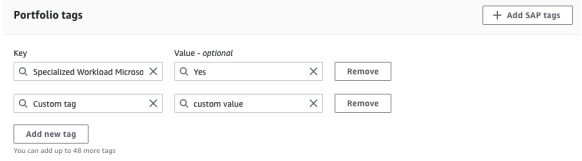
- To prevent accidentally sharing customer data with unauthorized people, sharing between AWS partners is only supported within the same APN organization. AWS partners can share with any AWS users though. Also, MPA doesn't support sharing the portfolios with customers since only APN Portal and AWS sing-on are supported for authentication at present.
- There are two permission types: *Full* and *Update,* the difference between them being that *Full* allows the new user to further share and also delete the portfolio, while *Update* doesn't.
- MPA doesn't currently track the changes made by the different people with access to the same portfolio. Please ensure you discuss with the ones you have shared the portfolio regarding any changes you or they are making.

Troubleshooting:

- The person I shared my portfolio with doesn't have access to it.
  Sometimes people have multiple email addresses within the same company. Please ensure the entered email address is the one they use for MPA access; in particular, validate the domain name and extension.

1.3. Edit portfolio properties

There are few attributes stored at portfolio level and which you can edit by going to *Actions* > *Edit properties* on the *All portfolios* page, or to the *Portfolio data* > *Properties* left nav link. On this page you can change the name of the portfolio, toggle whether the portfolio is used for testing purposes or for a real customer engagement, as well as add tags. Use the built-in or your own custom tags to help you label your portfolio as needed. The *Add SAP tags* button is a quick way to add SAP specific tags used by AWS consultants.



1.4. Copy a portfolio

Sometimes you may want to test different scenarios with subsets of the customer data, or just outside the capabilities provided through the assessments on a portfolio (see Assessment comparison). You can easily copy a portfolio from the *All portfolios* page, *Actions* > *Copy*. If the source portfolio is not a test portfolio, you will need to consider whether the new one should be, in which case please edit the portfolio to update its test indicator.

Note: MPA only copies the data as present in *Assets* and *Dependencies*, not the results. Also, all the assumptions are restored to default, so any custom values will need to be entered again (see Assumptions on how to use save/load).

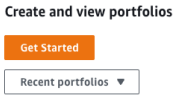
1.5. Delete portfolios

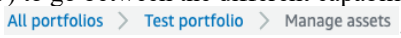
You may want to delete one or more portfolios, to keep your data neat or simply because they are no longer needed. You can do this from the *All portfolios* page, *Actions* > *Delete*.

Note: Please pay attention that this is a final action! Once deleted the data of that portfolio can't be recovered. If you are working on the same portfolio with other consultants or partners (you can easily see this by checking with whom the portfolio is shared), it is a good practice to consult with them before taking this action.

1.6. Navigating a portfolio

From the landing page, the *Get started* button opens the *All portfolios* page. Also on the landing page, if you have used any portfolios, you will see up to 5 of them in the *Recent portfolios* dropdown; this list is also available on the top right corner throughout MPA. On the *All portfolios* page use the portfolio name link in the list of portfolios to navigate to the desired portfolio.



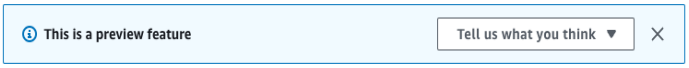
Within a portfolio, use the left navigation menu (left nav) to go between the different capabilities of MPA. You can also leverage the breadcrumbs bar at the top of the page 

.
The left nav is grouped as follows:

- *Portfolio data* – contains your assets and dependencies; here you can also visualize your data through dashboards, as well as change portfolio properties.
- *Import* – groups the two options for getting data into MPA: import from file and from ADS.
- *Collect* – find here the contact management and data collection capabilities.
- *TCO* – groups the higher level results of the TCO analyses: the summary of the TCO for regular and mainframe workloads, the per application cost summary and the standalone SQL Server optimization assessment.
- *AWS* – find here the TCO details for your regular workloads in AWS.
- *On-Premises* - find here the TCO details for your regular workloads On-Premises.
- *Migration strategy* – home to the server and application level R-type recommendations.
- *Migration cost* – find here the estimated migration cost for your regular workloads and a cash flow view of the two scenarios: AWS and On-Premises with migration and On-Premises without migration.
- *Plan* – application prioritization, dependency grouping and wave planning – all reside here.

Click the bell icon in the top navigation bar to see if there are any new messages from us, the MPA team. We use this to alert you about new features released or issues identified. The complete list of released features can be accessed also from the top nav bar: *Resources* > *What's new.*

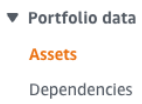
Use the generic *Feedback* > *Provide feedback*, for general feedback on the application; use the other options for raising a feature request or alerting us on an issue you came across. Some of our features are launched in *preview*, meaning that we are still iterating through them and actively seeking your inputs – do reach out and tell us what you think!



If you need to learn about the capabilities on a particular page, use the contextual help pane to the right (look for); it provides a brief synopsis with additional references. You can always open the MPA user guide found in the top nav bar under *Resources*, or refer to the FAQ page in the same place. We try to keep them up to date, but should you come across something wrong or outdated, don't hesitate to raise a bug through the feedback form.

2. Data entities

MPA works with two categories of data entities: assets and dependencies. The assets describe the On-Premises infrastructure components: servers, applications, databases, shared storage and network, while the dependencies describe the most important relations between them: server to application mapping, server communication, database to application mapping and application dependencies.



While both these data entities play many roles in the different MPA features, not all are needed for all the features. For TCO and migration cost estimation you need servers; if you have database data, as long as each database relates to a server, MPA will use those databases for TCO and migration cost estimation. MPA will use the applications, as long as server to application mapping and database to application mappings are provided, to estimate the per application TCO.
For application prioritization, you only need applications data. For dependency grouping and wave plans, MPA will use data from all the assets and dependencies.

Note:
- After changing any of the input data you need to press "Calculate" to regenerate the MPA assessments.
- You can obtain a migration cost estimate even without having the actual servers and databases info, you only need the count by R-type (see Estimate).
- Too chatty server communication data can result in a single creation of only one dependency group. No server communication data can result in too many groups, so you need to carefully review it for meaningful results.
- There is a common, special attribute to all entities in MPA: *In Scope of Portfolio*, in short *in-scope*. Use this to control which assets are considered "active" for the purposes of MPA results. This allows you to test out different scenarios with the data in the same portfolio, no need to create a separate one.

2.1. Servers

Server data is core to everything in MPA. MPA accepts on-premises attributes, as well as attributes describing the AWS preferences. While MPA can do with just a Sever ID specified (it applies default assumptions for all other missing data), please note below significant attributes to specify:
On-premises attributes:

Attribute	Relevant for	Default value*	Comments
Server ID	Throughout MPA	no	Mandatory. Doesn't need to be the actual ID, can be the server hostname or IP**. It has to be unique across all the servers.
OS Name and *OS Version*	• On-Premises TCO • Migration recommendations	*Linux*	Different source files might have the OS Name and version captured in the same field or separately. For this reason, MPA merges the values of *OS Name* and *OS Version* when using them.
CPU-Number of Processors	• On-Premises TCO • EC2 and RDS TCO • SQL Server optimization	*1*	When computing vCPUs, MPA multiplies the values of these 3 attributes. As best practices, for physical servers you will typically populate the number of processors with the number of sockets and the cores per processor with the cores per socket. For virtual machines (VMs), you can populate either number of processors or number of cores with the number of vCPUs of the VM. In most cases you would not need to worry about the number of threads.
CPU-Cores per Processor		*1*	
CPU-Number of Threads per CPU-Core		Per *Assumptions > AWS server recommendation > Default number of threads per CPU-Core*	

CPU-Peak Utilization	• On-Premises TCO • EC2 and RDS TCO • SQL Server optimization	Per *Assumptions > AWS server recommendation > Default peak CPU utilization*	Use toggle in assumptions to apply this override even if the server has valid value. For database servers, the override comes from *Assumptions > AWS database recommendation*
CPU-Average Utilization, RAM-Peak Utilization and RAM-Average Utilization – similar to above			
RAM-Total Size	• On-Premises TCO • EC2 and RDS TCO • SQL Server optimization	*0*	When importing, pay attention to the unit of measure (UOM).
Server-Usage	AWS TCO	Per *Assumptions > AWS server cost > Default production server usage* (if *Environment Type* evaluates to *Production*)	Also called *Server Uptime* It the *Environment Type* value evaluates to *non-production*, then the *Default non-production server usage* value is used instead.
Environment Type	• AWS TCO • Application-level cost estimate		When populated, values starting with *prod* or *prd* (case insensitive) evaluate to *Production*; anything else to *Non-Production*.
Environment Name	AWS TCO	*Production*	If the Environment Type is missing, MPA infers the environment type from the environment name: values starting with *prod* or *prd* (case insensitive) evaluate to *Production*; anything else (including missing value) to *Non-Production*.
Physical/Virtual	• On-Premises TCO • AWS TCO • Migration recommendations	*Physical*	Values (case insensitive) of *virtual, vm, false* and *no*, evaluate to *Virtual*.
Hypervisor	• On-Premises TCO • AWS TCO • Migration recommendations	*VMware*	Values not containing *xen, kvm* and *hyper* (case insensitive) or missing, evaluate to VMware.
Storage-Total Disk Size	• On-Premises TCO • AWS TCO	*1*	Pay attention to the UOM.
Storage-Utilization	AWS TCO	Per *Assumptions > AWS storage > Default Storage-Utilization*	
Storage-Snapshot Size	AWS TCO	Storage-Total Disk Size	When missing or 0, the value of the Storage-Total Disk Size is used instead.
Storage-Max Read IOPS/ Storage-Max Write IOPS	• On-Premises TCO • EC2 and RDS TCO	*0*	MPA adds them up for the max IOPS
Storage-Max Read IOPS Size/ Storage-Max Write IOPS Size	AWS TCO	*0*	MPA adds them up for the max throughput

CPU-Type	Used for Graviton recommendations and for on-prem Oracle license estimation	No	MPA assumes preference for Graviton recommendation if this attribute evaluates to *arm* (see EC2 instance preferences)
Datacenter ID	Presentation only	No	Use this to segment your servers as per customer needs.

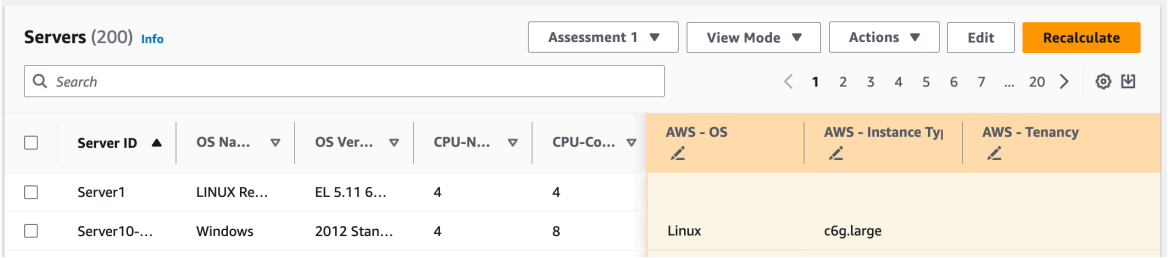
AWS target architecture preferences:

Attribute	Relevant for	Default value*	Comments
AWS - OS	AWS TCO	*OS Name* and *OS Version*	If specified, MPA uses this value for EC2 recommendations, else it uses the concatenated values of the *OS Name* and *OS Version*. You can specify any value, but if it doesn`t evaluate to Linux, Windows, RHEL or SLES, it will be interpreted as Linux.
AWS - Instance Type	AWS TCO	No	Specify only if you have a preference (see EC2 instance preferences), else leave it for MPA to recommend.
AWS - Tenancy	AWS TCO	Per *Assumptions > AWS server cost > Default tenancy*	Indicates if the EC2 tenancy should be *Shared, Dedicated Instance* or *Dedicated Host*.

* Default values apply for missing or, where applicable, invalid values.
** IPs are considered PII data (personal identifying information); if the customer has any PII or GDPR (General Data Protection Regulation) concerns, there is no reason to load IPs into MPA.
*** EC2 and RDS

Note:

- By default, in *Assets > Servers* presents only the view of the on-prem attributes. You can use the *View Mode* button to toggle between viewing the user specified target architecture attributes, the MPA inferred target architecture attributes (as generated following Calculate) or just the on-prem attributes.



- All on-premises attributes are stored at portfolio level, meaning they retain their values across different MPA assessments. Target attributes however can have different values between different assessments. Use the *Assessment* dropdown to toggle between them.
- MPA derives the meaning of input data by applying regex and default values. For example for *Hypervisor* MPA applies regex to identify if the value is either “HyperV”, “Xen” or “KVM”, else it assumes it is “VMware”.
- You do not need to import the data of physical hosts; import the virtual servers running on those hosts instead. If you do import servers with *ESX* as *OS Name*, MPA will automatically mark them as *Retire*, and they will be excluded from the AWS TCO calculation.
- Server *OS Name*, *CPU-Number of Processors* and *RAM-Total Size* are the minimum required to obtain TCO results; adding CPU and RAM utilization information improves the AWS estimation by tailoring the EC2 and RDS recommendations to the actual usage.
- Providing storage size (*Storage-Total Disk Size*) without the storage utilization can give and incorrect picture of your financial comparison.
- For an improved estimation of the On-Premises server cost, on a portfolio with several VMs, use the *Server-Target Host Type* attribute to specify the *Host type* for each VM. The *Host type* needs to match one of the values defined in *On-Premises > Server cost*, the *Host servers* tab, *Configure host types* button (see Compute).

2.2. Business applications

Application data in MPA is intended to represent the business applications; it is required for migration planning (application prioritization, dependency grouping and wave planning) and for application-level cost estimation. Before collecting the application data, identify the customer's business driver for migration so you can capture relevant application attributes. If a customer wants to migrate applications based on business criticality, then *Service Level Agreement, Business Criticality, Number of Users* can be used to identify the business criticality of an application. As for all other assets in MPA, you will need a unique ID for each application; provide an *Application Name* for a more meaningful identification.

Attribute	Relevant for	Override*	Comments
Application ID	Throughout MPA	No	Mandatory
Application Name	Presentation only	No	
Description	Presentation only	No	Meaningful short description of the application
Environment Type	Presentation only	No	For application-level cost estimate MPA infers the environments from the related servers.

*Overrides apply for missing or invalid values.

2.2.1. Additional attributes

For business applications only, MPA supports user-specified attributes to be used in data collection and application prioritization. You define new attributes, on the *Assets > Applications* page, click the *Add attribute* button; provide the attribute name (keep it short yet descriptive), the attribute type and data type. Note that MPA doesn't enforce the data type, this is for your reference only.
Should you decide you don't need any of the new attributes you created, you can use the *Delete attribute* button to remove them; all data associated with them will be deleted as well. Note that this is a final action and we can't recover the lost data.

2.3. Databases

Database information is used in On-Premises database TCO and AWS database recommendations, as well as for migration cost estimation. Gather On-Premises database details, including details of the servers each database resides on. The list below shows the most used database attributes; MPA supports more, feel free to upload their values for your reference and future capabilities.

Attribute	**Relevant for**	**Default value***	**Comments**
Database ID	Throughout MPA	No	Mandatory; unique across all databases.
Server ID	• On-Premises database TCO • RDS TCO	No	ID of the server on which the DB runs On-Premises; needs to exist in *Assets > Servers*; case sensitive.
DB Name	Presentation only	No	Name of the database
DB Instance Name	Presentation only	No	Name of the database instance
DB Instance Engine/Software	• On-Premises database TCO • RDS TCO	No	For RDS TCO this field is relevant only if the *Target AWS DB Service* is missing, and its value indicates a DB engine supported on RDS.
DB Instance Engine/Software Version	• On-Premises database TCO • License estimation for DH recommendations	No	Only meaningful for values that indicate *SQL Server* or *Oracle.*

DB Size	• On-Premises database TCO • RDS TCO	*0*	Needs to be less or equal to the *Storage-Total Disk Size* of the corresponding server; used for DB storage cost estimation On-Premises and in AWS.
Target AWS DB Service	RDS TCO	Per *Assumptions > AWS database recommendations > Default target database engine*	Supported values: *Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL, SQL Server* and *Oracle.*
Enterprise / Standard Edition	• On-Premises database TCO • RDS TCO	Per *Assumptions > AWS database recommendations > Database edition*	Supported values: *Enterprise, Standard, Express, Web, Standard One, Standard Two*.
License Model	• On-Premises database TCO • RDS TCO	Per *Assumptions > AWS database recommendations > Database license model*	Supported values: *Bring your own license, License included* and *No license required*.
DB Instance Deployment Option	• On-Premises database TCO • RDS TCO	Per *Assumptions > AWS database recommendations > Default deployment option*	Supported values: *Single-AZ* and *Multi-AZ.*

*Default values apply for missing or, where applicable, invalid values.
Note: *Maximum IOPS* is not currently in use, MPA leverages attributes from the corresponding server instead.

2.4. Shared storage

Shared storage data is used for On-Premises and in AWS cost estimation; import is not yet supported, but you can enter it manually in *Assets > Shared storage*.
For SAN storage, provide volume size and peak IOPS. IOPS information and volume size are used to recommend the EBS type. For NAS storage, provide volume size and select EBS or EFS (EFS is not available in all the AWS regions, hence validate the AWS region before selecting EFS). For Object storage, provide volume size and indicate if *Standard* or *Glacier*; infrequent access percentage applies to *Standard* only. Infrequent access refers to object storage that is not used on daily basis but used to keep data of inactive transactions. For EBS service, you can also specify the volume type to use; if not specified, MPA recommends one based on settings in Assumptions.

Attribute	**Relevant for**	**Default value***	**Comments**
Storage Description	Presentation only	"-"	Use this for your own convenience in identifying the shared storage items,
Storage Type	• On-Premises TCO • AWS TCO	No	Supported values: *SAN, NAS, Object*.
Disk Type	On-Premises TCO	For Max IOPS > 500 *SSD*, else *HDD*.	Supported values: *HDD* and *SSD.*
Raw/Usable	• On-Premises TCO • AWS TCO	No	Indicate *Usable* whenever OS, file share system and RAID overhead has been deducted from raw storage size.
Size	• On-Premises TCO • AWS TCO	No	Storage size.
EBS/EFS, Standard/Glacier	AWS TCO	No	AWS service to use.
EBS Volume Type	AWS TCO (EBS)	No	The volume type to use if EBS.
Max IOPS	AWS TCO (EBS)	No	Relevant for EBS only.
Infrequent Access	AWS TCO (S3)	No	Indicates the percentage of the Size to be used for IA; relevant for S3 only.

*Default values apply for missing or, where applicable, invalid values.

Note:
- Don't double count server disk storage with shared storage. Server disk storage that was ingested with server data needs not be included again as shared storage.
- MPA allows you to enter each On-Premises drive separately; if you only have the combined values, make sure you have the total IOPs as well.

2.5. Network

Network data is used for On-Premises and in AWS cost estimation (cost of *Data Transfer*); import is not yet supported, but you can enter it manually in *Assets* > *Network*.

Attribute	Relevant for	Comments
Size of Pipe	• On-Premises TCO • AWS TCO	The throughput in Mbits per second of the outbound connection.
Peak/Average Ratio		The ratio between peak traffic and average network traffic.
AWS Egress Data Transfer		The total amount of data in GB/Month expected to be transferred out of AWS in a month.

Note: You need to specify either the MPA *Size of Pipe* and *Peak/Average Ratio* values, or the *AWS Egress Data Transfer* value. If the latter is 0, MPA infers it from the former.

2.6. Dependencies

Dependencies information is used in planning (see Dependency groups and Migration wave) and enables estimation of the application-level cost.
Server to application information permits mapping of infrastructure to applications; one server may be used by more than one application and one application may use multiple servers. Application ID and Server ID are the two fields required; make sure corresponding servers and applications exist in *Assets*.
Server communication information lists the network traffic between servers. Source Server ID and Target Server ID are the two fields required; make sure corresponding servers exist in *Assets*.
Database to application information lists the databases the applications use; one database may be used by more than one application and one application may use multiple databases. Application ID and Database ID are the two fields required; make sure corresponding databases and applications exist in *Assets*.
Application dependency information lists the relations between applications. Source Application ID and Destination Application ID are the two fields required; make sure corresponding applications exist in *Assets*.

Note: You will see the *Application Mapped*, *Server Mapped* and *Database Mapped* fields. MPA is considering an application mapped to a server, if the application with that ID exists and is in-scope, and unmapped if the application either doesn't exist or is not in-scope.

2.7. Mainframe workloads

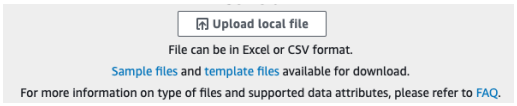
MPA offers an opinionated take to estimating the cost benefits of migrating mainframe workloads to AWS. You start by manually entering the details of the workloads in *Assets* > *Mainframes*. You will see the results under *TCO* > *Mainframe summary*, and you can modify corresponding assumption values from that page as well.

3. Data management

3.1. Import from file

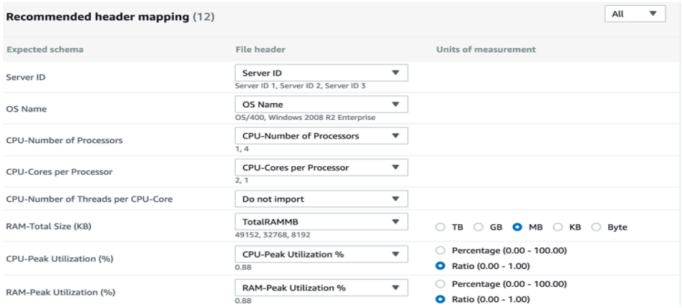
3.1.1. The import flow

MPA supports import by file for all entities, except *Shared storage* and *Network*.
While On-Premises data can come in many shapes and forms, MPA can understand Excel and CSV (comma-separated values) files; however, it can run into problems dealing with formatting like merged cells, formulas, special characters and locales. For this reason, we recommend to use the template or sample files provided (links in *Import* > *Import from file*).

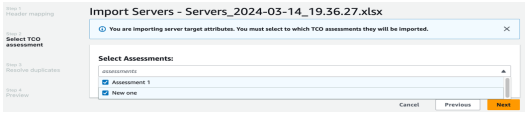


Currently, entities can be imported only one at a time; if you have them in Excel, make sure each entity is in its separate worksheet and upload that same file indicating the correct spreadsheet for each entity you want to import.

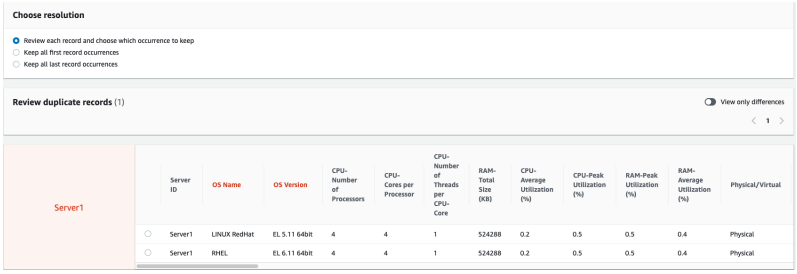
In the import process, MPA looks for headers matching preset patterns of the supported attributes and shows the *Expected schema* vs the *File header*. Please validate the mappings and make any necessary changes – this is particularly important with files that don't follow the MPA template.
Likewise, for numeric fields, MPA infers the unit of measure (UOM) or type (percentage vs ratio) from the file itself – please validate these values as well, the impact of a mismatch can be of orders of magnitude. MPA shows *Do not import* if it was not able to find a match; leave it as is or select the corresponding field. Note that the user experience is different between those attributes MPA identifies as *Recommended* versus the *Optional* ones: for the *Recommended* MPA presents the expected schema and allows you to select the file header, while for the *Optional* it works the other way around.



If you are importing any target attributes, as these are stored at assessment level, if you have more than one assessments in your portfolio, you are asked to specify the assessment(s) for which to import these.



On the next step in the import flow MPA shows any duplicates identified and require resolution. In red are shown the headers with different values between the records. MPA recognizes duplicates based on the entity id. Resolve the conflict by choosing a record or one of the built-in options at the top.



In the last step, MPA shows a preview of the data to be imported. Click Confirm Import to have the data ingested in MPA; you will be redirected to the corresponding *Asset/Dependency* page.

3.1.2. Things to consider

Before uploading a file to MPA, consider the following:

- Check for return characters in your file as they can result in erroneous import; in Excel, rows might seem empty, but can have hidden characters. More broadly, pay attention to elaborate formatting: merged cells, special characters can cause errors in the import process.
  Note: While MPA might not complain about special characters in the import flow, they might not be recognized when later processing the data. A known issue is that of handling double byte characters used in Japanese.

- Excel formulas can confuse the import process. Please remove invalid values such as "#N/A" or "#Error".

- Pay attention to not have alphanumeric characters for fields expecting number values. Also, the format in Excel of number fields can play tricks on you depending on your locale: for example, Excel might show "," but store "."; MPA reads what is stored, hence the ".", which can make a huge difference in the uploaded data.

- For those attributes with a unit of measure (UOM), you can specify the UOM in the column header (for example specify GB for storage fields); make sure the UOM matches the data in the column.

- MPA supports a maximum file size of 20 MB and a maximum of 30,000 records per file. While you might not have 30,000 servers to import, there are other reasons that can cause a file to be that big: graphs and pictures embedded in an Excel file, those hidden characters mentioned above (deleting the content of a row in Excel can leave Excel thinking that there is still data in there; best to delete the whole rows), or other worksheets not relevant to the import but that have lots of data.

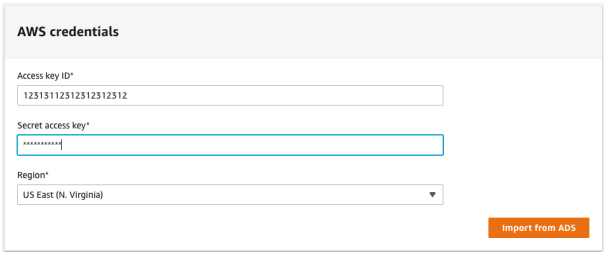
  Note: You can always repeat the import flow and upload more files for the same entity type. Just note that if MPA finds a record with the same id as one already imported, it will overwrite the existing one. The overwrite has a peculiarity: if an attribute is missing or has an empty value in the new file, MPA will keep the old value in the existing record! If that's not something you want to happen, you can always delete the existing records from MPA and upload the new file.

- There can be other causes leading to the import flow not being able to process your file. Common ones are password protected files and even certain firewalls. In case of failures in importing your file, remove the password if it has one, and try with the disabled firewall if one is present.

3.2. Import from ADS

For those customers that have used ADS (AWS Application Discovery Service) to collect data about their servers and applications, MPA allows to pull that info from ADS itself. MPA will load from ADS the details corresponding to *Servers*, *Applications* and *Server to application* dependencies.

You will need the secrets of a user authorized to access the customer's AWS account where the ADS data is stored. On the *Import* > *Import from ADS* page, enter the access key information and the region where the ADS data is stored in the customer's AWS account, and click *Import from ADS*.



Note:

- MPA is not part of the AWS Console.
- Deleting data from ADS will not delete it from MPA, and vice-versa.
- Importing duplicate data from ADS will overwrite the existing data in MPA.
- Importing duplicate data from a CSV or Excel file will overwrite the existing ADS data in MPA.
- Re-importing data from ADS will overwrite the existing records.
- You can't import from multiple AWS accounts. Create a new portfolio if you want to import data from a different AWS account.
- As a best practice, create an IAM user with only the *discovery:DescribeConfigurations* and *discovery:ListConfigurations* actions and use its access key.

3.3. Data collection

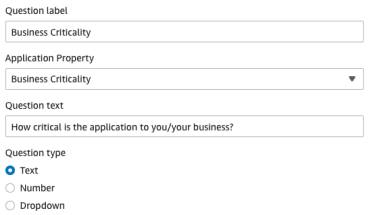
After importing assets in MPA, users can gather additional details about those assets from customer contacts. Typically, this is needed for getting a comprehensive picture about the customer's business applications.

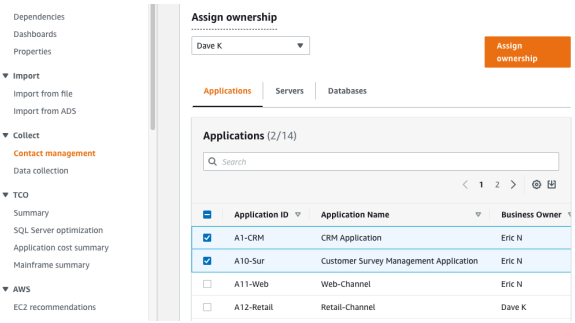
Data collection in MPA works at asset level, in that you gather data about one asset at a time by sending a questionnaire about that asset to a customer contact. You choose which asset needs more info, assign it to a customer contact, configure the asset's questionnaire, and then send that questionnaire.

You can manage contacts and assign them to assets on the *Collect > Contact management* page. MPA extracts customer contacts info from *Owner* attributes (*Email, Name* and *Phone number*) on the assets records. You can also enter the contact details manually: just click *Add contact* and provide user name and email address; phone number is not mandatory but, if entered, it is checked to be an internationally valid number.

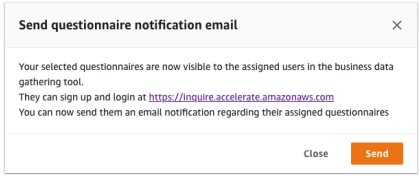
The questionnaire is the list of data points you want to inquire about, defined in the form of questions and associated asset attributes.

MPA provides template questionnaires specific to each asset type; you can modify these, either for all assets or individually per asset.
Use the *Edit questionnaire template* button on the *Collect > Data collection* page to customize the template questionnaire. Click the link associated with the asset id to customize the asset specific questionnaire.
In each approach you can edit and delete existing questions, and add new questions. A question consists of a brief description (the *Question label*), a detailed, user-friendly text to provide context to the customer contact (the *Question text*), question type and the attribute who's value the provided answer will update. The question type allows you to provide your own values, use the *Dropdown* type.
Make sure to click the Save button to persist your changes.



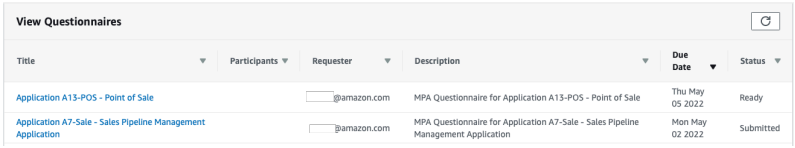


Before sending the questionnaire, make sure a contact is assigned to the corresponding asset. On the *Collect > Contact management* page, select the assets you want to assign, select the contact from the *Assign ownership* dropdown, then click *Assign ownership*.



After customizing the questionnaire and assigning a contact, you are ready to send it to the customer for feedback. On the *Collect > Data collection* page, select the assets you want to send, enter the *Due date* and click the *Send selected request(s)* button. Note that the due date is not enforced, but is still useful for setting an expectation with the customer. After pressing the *Send selected request(s)* button, you are prompted to send an email to the customer contact. Press *Send* and MPA will open your default email client with a new email populated with a predefined content for you to send to the customer. If you click *Cancel* on this prompt, the questionnaire is still assigned to the contact, but they will not receive a notification; naturally, the same applies if you don't send the email from your email client.

After receiving the email, the customer will need to create an account, if first time user, and login into the *Business Data Gathering* application (also known as *Migration Data Collector*, or MDC). After login, they will see the assigned questionnaires.



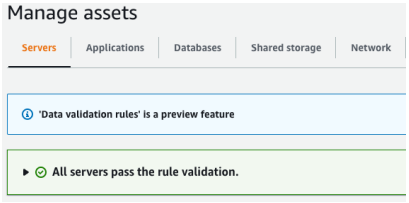
They will provide answers to the questions and have the option to either save those answers without sending them back to MPA, or submit them, at which point they will be reflected in MPA as well. Answers can be submitted more than once; the latest value will be reflected in MPA.

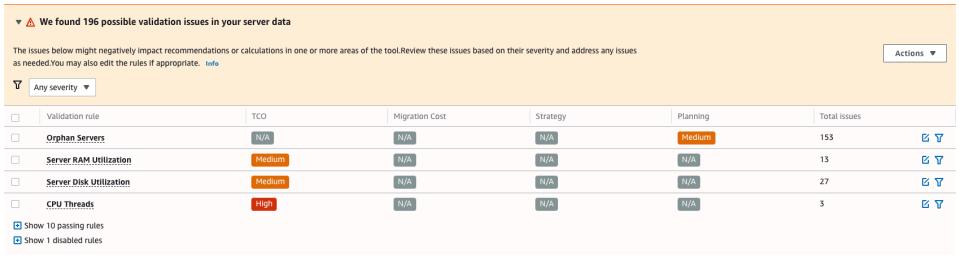
Note:

- Changes in MPA of a questionnaire already sent will reflect in MDC without the need to refresh.
- The "Participants" you see in the MDC screenshot does not apply to MPA functionality. MDC is used for one of our other applications where it comes into picture.
- The initial questionnaire status in MDC is different from the one in MPA. MDC will show *Ready* while MPA will show *Sent*. If the customer has made changes but not submitted them yet, both will show *In progress* and both will show *Submitted* if they submitted their answers.

3.4. Data validation

The accuracy of the MPA results depends on the quality of the imported data. While MPA doesn't validate the data at the time of the import (for example, in the file import flow MPA only checks for duplicate records), after the import has completed, on the page of the corresponding data entity, you will see the results of MPA validation. Think of this validation as flagging *potential* issues in the data; you review those issues and decide if to make any changes or not. This validation is based on built in rules, but you can create your own rules as well. To see the results per each rule, just expand the box. Use the funnel icon to filter to those records impacted by the rule. You can update the data in MPA (see Data update), or in the external file and re-import, after which you can trigger MPA to reassess the rules by clicking *Actions* > *Revalidate*.





To create your own validation rule, select *Actions* > *Add new rule*. If you have already created such custom rules in a different portfolio, you can import them in the current one: *Actions* > *Import rules*. You can edit a custom rule, but not built-in rules. You can disable/enable any rule. For a view of fields failing validation across all rules, select *Actions* > *Download report*. For details on the impact of a rule on the MPA features available for the impacted data, refer to the corresponding *Asset* tables above.

Note: When you define a rule, you specify the criteria for it to be *valid*; when executed it calls out what is *invalid*, the records that don't match the rule.

3.5. Data update

Once the data is imported into MPA, on the corresponding entity page, you can import more (*Actions* > *Import more records*), delete some or all of the records, bulk edit (using the Edit button) or in-line edit (hover over the value you want to change and click the icon). Note that deleting the value for some of the fields is not currently possible. Should you need to do that, the workaround is to download all the data in the table (use the icon on the top right of the table), modify the data as needed outside of MPA, delete the corresponding records in MPA and import them anew from the modified file.

For those attributes for which MPA expects specific values, when editing them in MPA, you will only be presented with those values. For this reason, for example you can't set *QA* or *Dev* for the server *Environment Type*, you can only select between *Production* and *Non-production*, which are the two values that MPA has logic around. If it is important for you set *QA/Dev* as the values, then the best approach is to follow the download all workaround described above. MPA will still interpret the updated values through its expectations, in this case handling *QA/Dev* as *Non-production*.

3.6. Data visualization

On the *Portfolio data* > *Dashboards* page, MPA allows you to visualize the data in your portfolio through built-in, customizable and create-your-own charts. Currently MPA only supports data of *Servers*. Use these charts to dive deep into your data and spot any anomalies/opportunities. For example, the built-in *Compute capacity (count)* and *Memory capacity (count)* charts can help you find out servers using excessive compute or memory and may be the candidate for refactoring.

You can add, edit and remove charts, and reorganize the dashboard layout.

Note: The *Dashboards* page relies on your browser's local storage. Therefore, it will not work if you have disabled use of Functional cookies. Also, if you switch to using a different browser or reset its local storage, all customizations will be lost.

4. Total cost of ownership

The total cost of ownership (TCO) reflects the financial benefits that the customer is expected to gain over time after migrating to AWS. MPA provides an estimation of what the customer currently pays based on the input data using industry benchmarks from the AWS Cloud Economics team.

For the AWS cost, MPA provides a directional TCO, in the sense that it is not expected to represent exactly the costs the customer will actually pay. This is because MPA only considers a core set of AWS services with limited settings, while the actual cloud implementation might consider more and many more configurations; in addition, at the time of the actual migration, customers might choose different AWS options, the On-Premises data might turn out to have been different and any number of other circumstances that make it hard to guarantee the cost beforehand.

4.1. AWS recommendation types

To estimate the AWS cost, MPA makes recommendations for the AWS architecture of your workloads and then prices them. By default, MPA recommends EC2 Shared/Dedicated Instance. You can also use Dedicated Host (DH), RDS and EC2 with SQL Server license included (LI) recommendations (see details in the corresponding sections of this guide). EC2 recommendations are determined based on the *AWS -Tenancy* attribute of the servers, or the *Default tenancy* in *Assumption > AWS server cost* when the server level value is missing. To enable EC2 with SQL Server LI and RDS recommendations use the corresponding toggles under *Assumptions > AWS database recommendations*.

A server might be eligible for more than one recommendation type. For this reason, MPA uses the following priority logic: EC2 with SQL Server LI has priority over RDS which has priority over EC2 (Shared/DI and DH).

Example: A database DB1 has target engine `SQL Server`, edition `Enterprise`, license model `license included`, is linked to a virtual server with Windows OS, tenancy Dedicated Host; both the server and the database are in scope.

- If EC2 with SQL Server is On (enabled in Assumptions), MPA gives it first priority even if the database or its corresponding server are eligible for other recommendations.
- If EC2 with SQL Server is Off, but RDS is On, MPA will assess an RDS recommendation.
- If all are Off, the server is assessed a Dedicated Host recommendation – with no consideration given to the database (it has license model `license included`).

Note: There are cases when a database qualifies for a certain recommendation type but MPA is not able provide a recommendation. You will see warnings in the EC2 recommendation and/or RDS recommendation pages. This is because certain attribute configurations are not supported by EC2 or RDS. For example number of cores or size of memory specified might be too high, or the instance type (when using an override) and the OS combination are not supported in the specified AWS region. Similarly, SQL Server BYOL is not supported on RDS.

For bring your own license (BYOL) scenarios, MPA can also estimate the Windows and database license costs. This applies to Dedicated Host recommendations and RDS with Oracle as target engine.

The results of these recommendations are the EC2 and/or RDS instance types and/or hosts, and based on these, the compute/database cost is determined. For each recommendation model, in addition to the compute/database cost, MPA also estimates the storage cost, network, IT labor and support cost.
For each cost category, On-Premises and in AWS, if you know of significant costs the customer is paying or expected to pay, you can enter them as adjustments (see Adjustments).

Note!

- MPA doesn't automatically rerun the TCO calculations after changes to the input data. If you change data (example: import more, delete some, change attribute values, change in-scope), make sure you press again the *Calculate* button for updated results.

Note:

- Just because a database is linked to a server and both are in scope it doesn't mean MPA will give a recommendation for that database. It has to qualify for either RDS, EC2 with SQL Server LI or DH recommendations and you need to have the corresponding toggles enabled.
- For equivalency, a database's cost is only estimated for On-Premises if it was estimated in AWS.
- Customer licensing situations can be complex; for this reason, by default, MPA doesn't estimate the database license cost (you can enable that in *Assumptions* > *Database licenses*); similarly, the cost of Windows licenses when a Dedicated Host recommendation is provided is also disabled by default (you can enable it in *Assumptions* > *Software and licenses*).

4.2. Assumptions

Both the On-Premises and in AWS TCO estimations are based on configurations called MPA assumptions. They control everything from the duration of the TCO, the AWS Region where the AWS services are deployed, to the preferred payment options and default values when missing input data. From any of the TCO pages, you can see and change these by clicking the *Modify assumptions* button. The assumptions are grouped based on the TCO area they cover, with a special group for *General* assumptions.

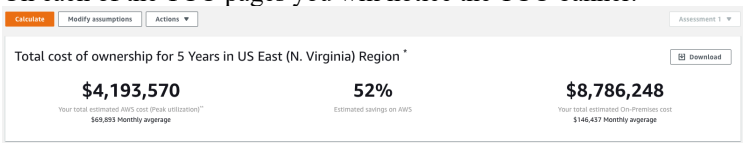
MPA uses industry standard values for the defaults of its assumptions based on data collected by the AWS Cloud Economics team. Every customer has different circumstances and the values should be changed to fit the customer's needs. Make sure to verify and validate with the customer if needed, the values of the MPA assumptions. You can modify the assumptions, save them and load them in different portfolios. Because saved assumptions are *objects* associated with your user, they are not visible to other users even if the portfolio is shared with them.

Example: You save the assumptions in "Portfolio A" as "EMEA assumptions" and share "Portfolio A" with someone else. They will not see "EMEA assumptions" as they try to load assumptions, but as long as you have loaded them in "Portfolio A", they will see their values.

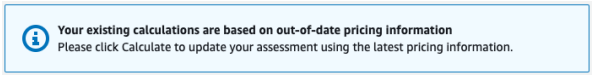
4.3. Navigating the TCO results

Use the left nav to switch between the different TCO results. After importing data, you would typically go to the *TCO* > *Summary* page. If this is a new portfolio and you have not run TCO before, MPA will allow you to change the assumptions; or you can click *Calculate* to proceed with the default values.
Under the *TCO* left nav group you will find other high-level pages, while under *AWS* and *On-Premises* groups, you will find details of the TCO summary broken by the respective cost categories.
In *TCO* > *Application cost summary*, you see the same costs as in *TCO* > *Summary* but distributed between the in-scope applications in your portfolio.
In *TCO* > *SQL Server optimization*, you get the potential SQL Server optimization if you have SQL Server databases in-scope in your portfolio and RDS recommendations is disabled. Note that the savings shown here are not included in the TCO summary.
In *TCO* > *Mainframe summary*, you find the mainframe specific estimations, if you have any mainframes in your portfolio.

On each of the TCO pages you will notice the TCO banner.



Use the *Modify assumptions* button to change any of the MPA assumptions and have the changes reflected in the TCO results. Under *Actions*, you have options to manage separate assessments (see Assessment comparison), and on the top right of the banner you see the currently selected assessment.
The main section of the banner shows the AWS and On-Premises estimated costs for the TCO duration, and as monthly averages, as well as the relative savings. Click the *Download* button to generate an Excel file with a summary of the TCO, a separate worksheet for the AWS details, another for an opinionated take to financial comparison and a worksheet with the assumptions used and their values.



The pricing rates MPA uses are synchronized from the AWS pricing service, hence should always reflect what you see in the public AWS pricing pages. If rates newer than the ones used last time the TCO results were calculated are available for any of the services MPA estimates, you will see this message above the TCO banner on the TCO pages.

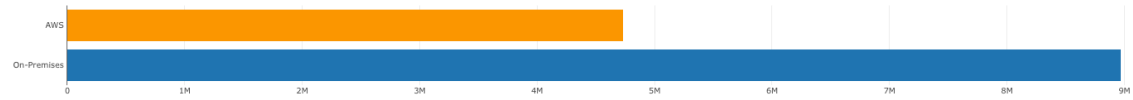
4.3.1. TCO summary

On the *TCO* > *Summary* page you have the *Summary report* of the TCO analysis for the data in-scope of your portfolio. It consists of:

- The AWS vs On-Premises cost comparison chart
- The AWS vs On-Premises cost breakdown chart and table

Summary report

AWS vs. On-Premises cost total



AWS vs. On-Premises cost breakdown

	Server	Database	Storage	Network	IT labor	Support	Total	Manual adjustment	Adjusted total
AWS	$2,256,224	$833,036	$983,102	$16,200	$345,112	$296,970	$4,730,644	$0	$4,730,644
On-Premises	$5,316,638	$586,156	$414,107	$1,526,510	$1,126,690	$0	$8,970,102	$0	$8,970,102

- The Cash flow summary comparing ‘as-is On-Premises’ and ‘complete migration to AWS’ scenarios.

Cash flow summary		Year 1	Year 2	Year 3	Year 4	Year 5	Residual and unused value
AWS	Server	$1,075,227	$142,845	$142,845	$1,075,227	$142,845	-$322,766
	Dedicated Host	$0	$0	$0	$0	$0	
	OS licenses - 3rd party	$0	$0	$0	$0	$0	
	vmware	$0	$0	$0	$0	$0	
	Database	$321,942	$88,940	$88,940	$321,942	$88,940	-$77,669
	Database licenses	$0	$0	$0	$0	$0	
	Storage	$196,620	$196,620	$196,620	$196,620	$196,620	
	Network	$3,240	$3,240	$3,240	$3,240	$3,240	
	IT labor	$69,022	$69,022	$69,022	$69,022	$69,022	
	Support	$102,078	$33,013	$33,013	$102,078	$26,787	
	Total	**$1,768,131**	**$533,681**	**$533,681**	**$1,768,131**	**$527,455**	**-$400,435**
On-Premises	Server	$1,227,474	$0	$0	$0	$0	
	Server maintenance	$184,121	$184,121	$184,121	$184,121	$184,121	
	Server rack/hardware	$362,804	$0	$0	$0	$0	
	Server facilities	$310,665	$310,665	$310,665	$310,665	$310,665	
	Server software	$1,252,432	$0	$0	$0	$0	
	Database	$368,227	$54,482	$54,482	$54,482	$54,482	
	Database licenses	$0	$0	$0	$0	$0	
	Storage	$206,917	$0	$0	$0	$0	
	Storage rack/hardware	$3,700	$0	$0	$0	$0	
	Storage facilities	$33,600	$33,600	$33,600	$33,600	$33,600	
	Storage software	$35,490	$0	$0	$0	$0	
	Network	$528,061	$249,612	$249,612	$249,612	$249,612	
	IT labor	$225,338	$225,338	$225,338	$225,338	$225,338	
	Total	**$4,738,829**	**$1,057,818**	**$1,057,818**	**$1,057,818**	**$1,057,818**	**$0**
Gain / Loss		**$2,970,699**	**$524,137**	**$524,137**	**-$710,313**	**$530,363**	**$400,435**

Use the *Download* button in the TCO banner to get an Excel version of this report (see more in TCO download).

4.4. EC2 recommendations

When it comes to TCO, *EC2 recommendations* is the default mode of analyzing your data for AWS cost. MPA is taking all the in-scope servers in your portfolio and finds the most cost-efficient instance type and payment option to rehost them on AWS EC2. You don't have to enable EC2 recommendations, it is what MPA does by default.

Use the *AWS - Tenancy* attribute on the server (*Assets > Servers*) to designate whether *Shared, Dedicated (Instance) or Dedicated Host* EC2 recommendations should apply. If this value is missing, MPA uses the value specified for *Default tenancy* in *Assumptions > AWS server cost*. If you have enabled RDS, then it takes priority, and the servers that qualify for those recommendations are not considered for EC2 recommendations.

4.4.1. EC2 sizing

Identifying the recommended instance type is also referred to as "sizing". You can see the sizing results in *AWS > EC2 recommendations*. MPA uses the server specs (vCPUs and memory) and their utilization (average, peak or on-prem capacity as specified in *Assumptions > AWS server cost > Server recommendation*) to find the right-sized instance type for the On-Premises server.

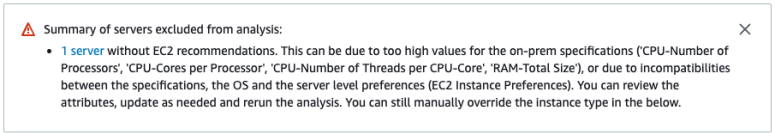
The following attributes defined on the server in *Assets > Servers* are used in sizing;

- *CPU-Number of Processors, CPU-Cores per Processor, CPU-Number of Threads per CPU-Core* and *CPU-Average Utilization* and *CPU-Peak Utilization* to determine the needed compute power: CPUs times cores times threads times utilization %.
- *RAM-Total Size*, *RAM-Average Utilization* and *RAM-Peak Utilization* to determine the needed memory: ram times utilization %.
- *Physical/Virtual*; for physical servers, the compute capacity can be multiplied with a factor of 2, to compensate for the fact that physical processors can be twice as powerful as virtual CPUs (see *Physical to EC2 instance factor* below).
- *AWS - OS*; indicates the OS to use for the EC2 instance. If not specified, MPA uses the on-prem value by concatenating *OS Name* and *OS Version*.
- *AWS - Tenancy*; the values *Shared* and *Dedicated* (Instance) indicate EC2 recommendations, while *Dedicated Host* is used for DH recommendations.

The following assumptions are also considered:

- *General > AWS Region*; only instance types available in the region are considered.
- *AWS server cost > Server recommendation*; indicates which utilization type to use for compute and memory calculation.
- *AWS server cost > Default tenancy*; if tenancy is not specified at server level.
- *AWS server recommendation > CPU buffer* and *RAM buffer*; used as multipliers for the final compute and memory needs (for average and peak utilization types).
- *AWS server recommendation > Default average CPU utilization* and *Default average Peak RAM utilization;* used when a value is not specified for the corresponding average attributes; you can choose to use these values even when server level values are provided by enabling the corresponding toggle.
- *AWS server recommendation > Default peak CPU utilization* and *Default peak RAM utilization*; used when a value is not specified for the corresponding peak attributes; you can choose to use these values even when server level values are provided.
- *AWS server recommendation > Physical to EC2 instance factor*; when set, MPA multiplies with the factor of 2 the compute specs for physical servers (servers for which the value of the *Physical/Virtual* attribute evaluates to *Physical*). Note that the default value for *Physical/Virtual* is *Physical*.
- *AWS server recommendation > Do Not Recommend These EC2 Instance Types*; specifies a list of instance types for MPA to exclude (see also server level EC2 instance preferences below).

On the same page, *AWS > EC2 recommendations*, MPA will provide a warning box in case it was not able to generate recommendations for any of the



in-scope servers. Click the link to be redirected to the *Assets* > *Servers* page and see the servers list filtered to those that ran into this issue.
Currently we are not able to always provide you with the exact reasons for why a recommendation was not possible, hence we ask that you validate the data. Common situations that can appear:

- Too high vCPU specifications: check *CPU-Number of Processors, CPU-Cores per Processor, CPU-Number of Threads per CPU-Core* and *CPU-Average Utilization* and *CPU-Peak Utilization* for abnormal values (invalid or too high).
- Too high memory specifications: check *RAM-Total Size*, *RAM-Average Utilization* and *RAM-Peak Utilization* for abnormal values (invalid or too high). Also check that the memory's unit of measure (KB, MB, GB) is accurate for the provided value in import.
- Values for utilization attributes are set to 0. Check the *CPU-Average Utilization*, *CPU-Peak Utilization, RAM-Average Utilization* and *RAM-Peak Utilization* on server level. If their values are missing, check the corresponding values in *Assumptions* > *AWS server recommendation*. Also, check that the utilization's unit of measure (Percentage, Decimal) is accurate for the provided value.
- OS and AWS Region: sometimes the OS, CPU and memory specs can result in an instance type that is not supported in the AWS Region you have specified. To check this scenario, you can change the *AWS - OS* to *Linux*, as most regions do support *Linux* configurations while not all support *Windows, SUSE or RHEL* for the same tenancy, vCPU and memory specifications.
- The provided server level instance type override is invalid or not supported in the specified AWS region. Check the EC2 instance preference attribute (see below) and the exclusions list in *Assumptions* > *AWS server recommendation.*
- The specified server level instance preference results in no matches. Check the EC2 instance preference attribute. This can also happen when portfolio level exclusion preferences are used, as one can negate the other (see below).

4.4.2. Server level EC2 instance preferences

You can specify preferences for EC2 instances at server level using the *AWS - Instance Type* attribute in *Assets* > *Servers*. This can be in the form of a complete instance type, or generic preferences. When a complete, valid instance type is provided (ex: *c6i.xlarge, m5a.2xlarge*), if it is available in the specified AWS region, it will be used as an override for all CPU utilization options.

For the generic preferences, you can use "amd", "intel", "burst" and "not" as keywords. You can specify multiple preferences separated with "," in which case, MPA applies an AND relation to any negative preferences and OR relation to positive preferences. For example:

- Preference for Intel instances: *intel*
- Preference for burstable instances: *burst*
- Preference for AMD, not burstable instances: *amd, not burst*
- Preference for c7, but not amd: *c7*, not amd*
- Preference for either c7 or c6: *c7*, c6**

If you need more granular preferences, these wildcards are supported: "*" (matches zero or more alphanumeric characters) and "+" (matches one or more alphanumeric characters); for example:

- Preference for Inferentia instances: *inf**
- Preference for network optimized instances: *+n**
- Excluding micro instances: *not *.micro*

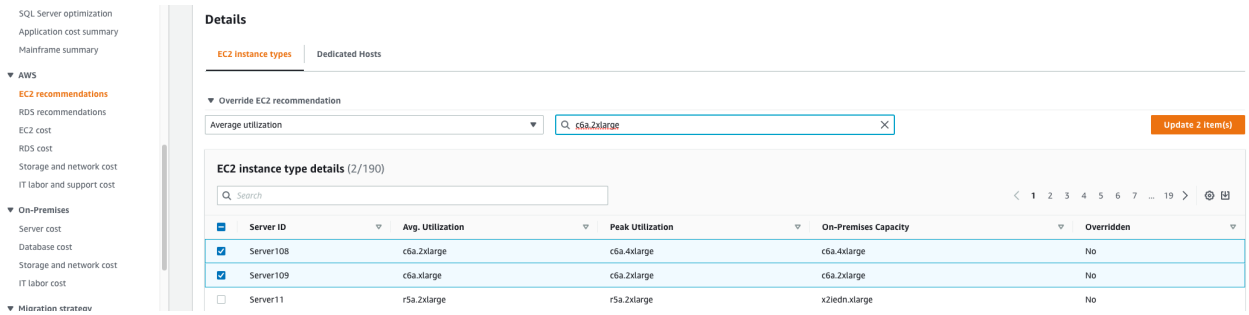
If you need to apply same EC2 preferences to all the servers in the portfolio, you can specify *exclusions* in *Assumptions* > *AWS server recommendation* > *Do not recommend these EC2 instance types*. This setting works similarly to the server level preferences, with a major difference: at server level you specify what you prefer to be recommended, here you specify what you prefer *not* to be recommended.
Note that, if server level preferences are also specified, the recommendation is based on the intersection of server level and portfolio level preferences. For this reason, pay attention that portfolio level preferences don't reverse server level, in which case there will be no recommendation made!

For example: a server level preference for burstable instances *burst* and a portfolio level exclusion of *burst*, will cancel each other and result in no recommendation being made for that server.

Graviton instance types are based on 64-bit Arm cores and therefore *not suitable for all workloads*. MPA does look for Graviton recommendations if the *CPU-Type* attribute of the server evaluates to *arm*. However, if the *CPU-Type* is not specified or not *arm*, for those workloads that you know can run on Arm processors, you can specify Graviton as the preference by entering "graviton" in the *AWS - Instance Type*.
Note: Graviton might not be available for the specified AWS region, OS and needed vCPUs and memory. If you requested Graviton and an instance type it is not provided after *Calculate*, remove the Graviton preference to check that indeed this preference caused the missing recommendation.

4.4.3. EC2 recommendations manual override

If you need to update the EC2 instance type recommendation that MPA made, you can do this on the *AWS > EC2 recommendations* page, by using the *Override EC2 recommendation* capability: select the servers and the utilization method to override, specify the instance type to consider, then click the *Update* button. MPA will make the update and rerun the TCO assessment, updating also the EC2 cost.



You can use the same mechanism to undo an override, just select *Undo Manual Override* instead of specifying an instance type.

Note: When the override value was specified at server level, in the *AWS - Instance Type* attribute in *Assets > Servers*, undo will not work since the server level preference takes precedence; you will need to remove that preference instead.

4.4.4. EC2 cost estimation

MPA makes an initial hint at pricing in the *AWS > EC2 recommendations* page, through the *Recommended payment option*; MPA identifies the most cost-effective payment option for that instance/host type, from the options selected in assumptions, under *AWS server cost > EC2 payment types*. If you are wondering how come *On-Demand* comes up as a recommended payment option, this is because MPA, for *On-Demand* only, adjusts the cost with the server uptime (the *Server-Usage* attribute in *Assets > Servers*); hence, for servers with low uptime, it can be more cost effective to use *On-Demand*.

Note: If the *Server-Usage* attribute is not specified, MPA uses the default value from *Assumptions > AWS server cost* corresponding to the server's environment.

MPA calculates the *Residual* value if the TCO duration (as set in *Assumptions > General >Analysis duration*) is not a multiple of 3. The *Residual* is the upfront cost for 3 year Reserved Instances (RI) that would be paid but not earned, due to the TCO duration exceeding the RI renewal.
Example: TCO for 5 years, *Standard 3yr - All Upfront* cost of $6,498; the *Residual* is $6,498 * (6-5)/ 3, because the customer would not make use of 1 year worth of the 2nd all upfront payment.

MPA also estimates the *Reserved Instance Volume Discount*, a tier-based discount that AWS provides based on the amount spent on reserved instances (see also the public documentation on *RI Volume Discount*: here and here).

For more details on the estimated compute cost in EC2, see the *AWS > EC2 cost* page. It presents summary level information, as well as details grouped by EC2 instance types, server level, payment option, and cash flows

The *EC2 instance types* tab shows the total EC2 cost broken down by the recommended instance/host type. The costs are computed for the TCO duration and include the residual, instance volume discount and host unused value.
The *EC2 cash flows* tab shows the total EC2 cost at server level, distributed across the TCO duration.
The *EC2 pricing detail* tab shows the EC2 cost at server level for each of the selected payment options (in *Assumptions > AWS server cost*); the cost for the *Recommended payment option* includes the residual, instance volume discount and host unused value.
The *EC2 payment option summary* tab shows the total EC2 cost for all the servers in scope for each of the selected payment options – this is a hypothetical cost, should each of those options be chosen for the whole portfolio.
The *EC2 payment option detail* tab enhances upon the *EC2 payment option summary* by showing the breakdown by instance/host type and OS, of the EC2 costs for a selected payment option.

4.5. Dedicated Host recommendations

Dedicated Hosts are required for those customers with stringent software license requirements (example: Windows, SQL Server), and/or for regulatory and security compliance. MPA currently doesn't assess the *need* for a dedicated host, that is left for the user to decide and indicate so through the *AWS - Tenancy* attribute, the *Dedicated Host* value. You can set this per server, or at portfolio level, in *Assumptions > AWS server cost*, if you want it applied to all the servers in your portfolio, when server level value is missing.

The costs associated with compute in Dedicated Hosts are not limited to the cost of the hosts themselves, license cost is an important aspect. Since license circumstances differ greatly between customers, MPA has these disabled by default. Enable an estimation of the BYOL license cost for Windows in *Assumptions > Software and licenses*, and an estimation of the BYOL license cost for SQL Server in *Assumptions > Database licenses* (this applies only if RDS recommendations are turned off, see the priority order and example in Total cost of ownership (TCO)).

4.5.1. Dedicated Host sizing

MPA determines how to optimally "pack" the in-scope servers with *Dedicated Host* tenancy on AWS hosts, starting from the on-prem specs (vCPUs and memory) and accounting for the on-prem utilization (average, peak or on-prem capacity as specified in *Assumptions > AWS server cost > Server recommendation*); for physical servers, the *Physical to EC2 instance factor* toggle in *Assumptions > AWS server recommendation* is also considered. Based on the *AWS - OS* (if specified, else concatenated value of *OS Name* and *OS Version*) of the servers, MPA infers the Windows license needs, if applicable. Similarly, if the server is linked to in-scope SQL Server databases, based on the *DB Instance Engine/Software*, *DB Instance Engine/Software Version* and the *Enterprise/Standard Edition* attributes of the database, MPA determines the SQL Server needs. The packing algorithm attempts to optimize for license cost as well.

Note: The "packing" algorithm might not provide cost optimal results for the last host, so if you have a portfolio that requires very few hosts, or just one, then the recommendation MPA provides might not be the best! If you find this is the case, you can switch to using *Shared/Dedicated Instance*.

Note: EC2 instance type preferences and instance type exclusions (in *Assumptions > AWS server recommendation*) do not apply to Dedicated Host recommendations. Due to the nature of the packing logic, override of the host type and of the instance type for an instance within that host is not currently supported. There is a way to override the host type for a *single server*: if a valid host type is specified as the *AWS - Instance Type* attribute of the server, MPA honors the provide value and doesn't attempt "packing" for that server. Use this with caution, usually for servers requiring a large number of vCPUs or large memory, like SAP workloads.

You can see the sizing results in *AWS > EC2 recommendations*, the *Dedicated Hosts* tab. The summary view lists all the hosts needed, their type and specifications, as well as the host's number of instances of each size and the number of unallocated vCPUs. The server level view lists all the servers assessed for Dedicated Host recommendation with their recommended host and instance type.

4.5.2. Dedicated Host cost estimation

MPA estimates the cost of the recommended Dedicated Hosts, as well as 3rd party OS and DB license costs. The Dedicated Host cost estimation accounts for the cost of each host as per AWS pricing for the selected region, as well as for residual amount (if a Reserved Instance pricing option is recommended and TCO duration in years is not a multiple of 3), and unused cost. You can toggle subtracting the unused cost in *Assumptions > AWS server cost > Apply unused capacity* (default is *On*).

If Dedicated Host recommendations were made, you will see two more tabs on the *AWS > EC2 cost* page:

- The *Dedicated Host details* tab shows the costs of each host for the TCO duration and the recommended payment option. The costs include the unused value and residual.
- The *Software and licenses* tab shows estimation of the Windows BYOL license cost on DH (if *Estimate Windows licenses on Dedicated Host* toggle is enabled in *Assumptions > Software and licenses*) and the SQL Server BYOL license cost on DH (if *Include database license assessment in TCO* toggle is enabled in *Assumptions > Database licenses*).

In addition, the *EC2 instance types* tab gives a view of the host instance families alongside the shared/dedicated instance types.
The *EC2 cash flows* and *EC2 pricing detail*, present the costs at server level. For this reason, the Dedicated Host cost is distributed across all serves with a Dedicated Host recommendation. The cost assigned to a server subtracts the residual amount and the unused cost (if enabled in *Assumptions > AWS server cost*); in other words, the total cost of all servers with tenancy *Dedicated Host* is equal to the total Dedicated Host cost shown in summary, minus residual, minus unused cost.

4.5.3. Dedicated Host - 3rd party license estimation

You can also enable estimating the OS license, in *Assumptions > Software and licenses*. Since customers will have different situations when it comes to OS licenses, this option is turned off by default. If enabled, MPA estimates the OS license cost for Windows servers only, based on the *AWS - OS* (if specified, else *OS Name + OS Version*) provided at server level; it attempts to map against known Windows versions (Windows Server Standard/Datacenter 2008, 2012, 2016 and 2019), else it defaults to *Windows Datacenter 2016*. The algorithm MPA uses to determine the hosts needed, also estimates the number of licenses required for the specified OS versions. MPA uses that number along with the Windows 2-cores license cost and annual support ratio defined in *Assumptions > Software and licenses* to estimate the one-time license and yearly support costs; to the cost calculated in this way, MPA also applies the software discount specified in *Assumptions > On-Premises discount/adjustment* (the assumption is that this carries over from On-Premises to the cloud).

Note: Currently MPA doesn't support Dedicated Host with *License Included* option (i.e., using an Amazon Windows AMI); for that reason, for DH tenancy, MPA assumes all servers running Windows to be BYOL.

Similarly, MPA can estimate the DB license cost for SQL Server BYOL. This too is controlled by a toggle, in *Assumptions > Database licenses*. In addition, for MPA to know to which databases the license estimation applies, few other criteria are needed: the tenancy is *Dedicated Host* for the server mapped to the database, the *Target AWS DB Service* is *SQL Server*, the *License Model* is *Bring your own license* on the database record (*Assets > Databases*). When these are met, MPA uses the *DB Instance Engine/Software, DB Instance Engine/Software Version*, and the *Enterprise/Standard Edition* from the database record to determine the SQL Server edition and version (SQL Server Standard/Enterprise 2008, 2012, 2014, 2016, 2017 and 2019), with the default being *SQL Server Enterprise 2019*.

Note:

- As long as the other conditions are met, MPA assumes the default of *SQL Server Enterprise 2019* even if the *DB Instance Engine/Software, DB Instance Engine/Software Version* are different than SQL Server, as long as the server tenancy is *Dedicated Host,* the *Target AWS DB Service* is *SQL Server*, and the *License Model* is

Bring your own license; they in fact should point to a SQL Server valid version as the customer is supposedly retaining their license given the BYOL license mode.
- While MPA has the price for only one version (Windows 2016 Datacenter - licensed per number of physical cores), it does actually identify the license needs per host for the *specific* Windows versions, some of which are licensed per number of sockets, hence less demanding as number of licenses, though the license cost itself might be higher.

4.6. Recommendations for EC2 with SQL Server license included

For customers that want to run their SQL Server databases in EC2 with license included (LI), you can enable the corresponding toggle in *Assumptions > AWS database recommendation*. This recommendation type takes priority over any other recommendations the database or its corresponding server might qualify.
For a database to qualify, besides enabling the toggle, you need to ensure the following:
- The database is in scope (`In Scope of Portfolio` attribute of the database).
- The target engine for the database is "SQL Server"; this means "SQL Server" is indicated either by the `Target AWS DB Service` attribute of the database, or (if that is missing/empty) by the source engine (the `DB Instance Engine/Software` attribute of the database), or (if that too is missing/empty) by the default value in *Assumptions > AWS database recommendation*.
- The edition is one of "Enterprise", "Standard" or "Web"; this can be either the `Enterprise / Standard Edition` attribute of the database, or (if that one is missing/empty) the default value in *Assumptions > AWS database recommendation*.
- The license model is "License included"; this can be either the `License Model` attribute of the database, or (if that one is missing/empty) the default value in *Assumptions > AWS database recommendation*.
- The corresponding server (as specified through the `Server ID` attribute of the database) exists, is in scope and has an OS that evaluates to "Windows", "RHEL" or "Linux".

Note: MPA does a best effort attempt to match the values provided against those expected: it is case insensitive and recognizes some equivalences (like RedHat for RHEL). Also, since default values in assumptions apply to all the corresponding records in your portfolio, to avoid conflict of intentions between different databases, it is best that you provide the intended values for each database.

If the database doesn't qualify for EC2 with SQL Server LI recommendation, MPA will attempt to find a match based on the recommendation type with next priority (see also AWS recommendation types).
In some cases MPA will qualify the database for EC2 with SQL Server LI recommendation, but not be able to generate one. This is usually due to too high specifications: check the vCPUs, memory and utilization attributes of the corresponding server; availability for your specified AWS region can also be a problem: check if an offer exists for that region, specified OS, SQL Server edition and (when instance override is used) instance type in the public AWS pricing website.

For the databases MPA is able to make an EC2 with SQL Server LI recommendation, it estimates the cost based on the rate of the instance type, OS and SQL Server edition. Find the recommendations and corresponding costs on the same pages as EC2 Shared/Dedicated Instance results (*AWS > EC2 recommendations* and *AWS > EC2 cost*); look for the `Pre-installed software` field to be populated with the value corresponding to the edition of the database. If more than one database are linked to the same server, MPA provides a single EC2 with SQL Server LI recommendation for all; if they have different editions, it considers `Enterprise`, `Standard` and `Web` in this order.

Note:
- The instance cost includes the license cost for SQL Server, similar to how it includes the license cost for the OS (when other than `Linux`); MPA doesn't separate these from the cost of the "base" instance.
- Instance override works for EC2 with SQL Server License included, similar to EC2 Shared/Dedicated Instance recommendations: specify your override either at server level (the *AWS - Instance Type* attribute) or through the *EC2 recommendations* page; MPA considers your preference along with the corresponding SQL Server edition as pre-installed software; if the instance type + pre-installed value is invalid, you will see an error.

For storage, MPA makes an EBS recommendation based on the selected volume types in *Assumptions > AWS storage*; for this MPA uses the storage size of the server (the `Storage-Total Disk Size` attribute) and applies all other assumptions that apply to server storage and EBS recommendations.

For On-Premises cost estimates, MPA treats the database and its corresponding server that qualifies for EC2 with SQL Server LI as an On-Premises database; find the results on the *On-Premises > Database cost* page.

4.7. Recommendations for EC2 shared with Oracle bring your own license

For customers that want to run their Oracle databases in EC2 shared/dedicated instance with bring your own license (BYOL), MPA can estimate the Oracle license cost. For this you will need to enable the `Enable EC2 recommendations` toggle in *Assumptions > AWS database recommendation* as well as the `Include database license assessment in TCO` toggle in *Assumptions > Database licenses*.

Note: This recommendation type takes priority over any other recommendations the database or its corresponding server might qualify. For example, if you also have RDS enabled, MPA will not estimate the RDS cost for this database. Likewise, even if the server's tenancy is "Dedicated Host", MPA will give priority to EC2 shared with Oracle bring your own license.

For a database to qualify, besides enabling the toggles, you need to ensure the following:

- The database is in scope (`In Scope of Portfolio` attribute of the database).
- The target engine for the database is "Oracle"; this means "Oracle" is indicated either by the `Target AWS DB Service` attribute of the database, or (if that is missing/empty) by the source engine (the `DB Instance Engine/Software` attribute of the database), or (if that too is missing/empty) by the default value in *Assumptions > AWS database recommendation*.
- The edition is one of "Enterprise", "Standard", "Standard One" or "Standard Two"; this can be either the `Enterprise / Standard Edition` attribute of the database, or (if that one is missing/empty) the default value in *Assumptions > AWS database recommendation*.
- The license model is "Bring your own license"; this can be either the `License Model` attribute of the database, or (if that one is missing/empty) the default value in *Assumptions > AWS database recommendation*.

If the database doesn't qualify for EC2 with SQL Server LI recommendation, MPA will attempt to find a match based on the recommendation type with next priority (see also AWS recommendation types).

For the databases MPA makes an EC2 shared with Oracle BYOL recommendation, it presents the license cost aggregated by edition *AWS > EC2 cost > Software and licenses* page. MPA estimates the license cost by computing the one-time cost and the yearly support cost based on the number of licenses required for the instance type recommended. The number of licenses is the number of vCPUs of the instance type divided by 2 (for "Enterprise" edition) or by 4 (for the standard editions), rounded up to the nearest integer. The one-time cost is calculated as number of licenses times the cost corresponding to the edition, as configured in *Assumptions > Database licenses*, times the corresponding discount. The yearly support cost is calculated as the license cost times the corresponding support percentage, also configured in *Assumptions > Database licenses*.
For databases with EC2 shared with Oracle BYOL recommendation MPA also computes the corresponding on-premises database and license costs (see also Database licensing estimation).

Note:

- The yearly support cost MPA presents is for the whole TCO duration.
- Instance override works for EC2 shared with Oracle BYOL, similar to EC2 Shared/Dedicated Instance recommendations: specify your override either at server level (the *AWS - Instance Type* attribute) or through the *EC2 recommendations* page.
- Besides license cost, Oracle BYOL has no impact on the instance cost – it is the same as if the server would be estimated a regular EC2 shared/dedicated host recommendation.

- License costs have a high variability based on the customer agreements with the vendors. Always check with the customer what license cost, discount and support percentage to use in *Assumptions > Database licenses*!

4.8. RDS recommendations

For databases, MPA can perform a database specific TCO assessment, that is: estimate the On-Premises and on RDS costs for the in-scope databases in your portfolio. MPA provides the estimated cost on RDS for the various payment options available; the storage cost for those databases on RDS; the estimated number of license and cost; support and IT labor cost. Likewise, for the On-Premises cost MPA estimates the database cost, storage cost, license, support and IT labor cost. You can turn on the database assessments functionality in *Assumptions > AWS database recommendation > Enable database recommendations*. When enabled, all servers linked to databases (through the *Server ID* attribute of the database, case sensitive) will be treated as database servers and used only to determine RDS and On-Premises database costs. If a database is not in scope but its corresponding server is in scope, MPA will consider that server as regular server and determine its EC2 recommendation and AWS and On-Premises costs.

Note: If both EC2 with SQL Server LI and RDS recommendations are enabled, EC2 with SQL Server LI has priority. For more on recommendation priority see AWS recommendation types.

4.8.1. RDS sizing

As for EC2 recommendations, you can specify the utilization method to consider (there is separate assumption from servers under *Assumptions > AWS database cost*, just in case you want to consider different methods for databases).

MPA uses the following attributes to find the best match among the RDS supported offers for your AWS region:

- From the corresponding server:
  - *CPU-Number of Processors, CPU-Cores per Processor, CPU-Number of Threads per CPU-Core* and *CPU-Average Utilization* and *CPU-Peak Utilization* to determine the needed compute power: CPUs times cores times threads.
  - *RAM-Total Size*, *RAM-Average Utilization* and *RAM-Peak Utilization* to determine the needed memory.
- From the database record itself:
  - *Storage-Total Disk Size* to determine the needed storage.
  - *Target AWS DB Service*
  - *DB Instance Deployment Option*
  - *Enterprise / Standard Edition*
  - *License Model*

MPA is making a best effort attempt to match the provided values to the expected ones for the target engine, DB edition, license model and deployment option. If a valid value is not identified, then the default values from *Assumptions > AWS database recommendation* are used. A special case is that of the target engine where, if a valid input value was not identified, MPA attempts first to use the corresponding *DB Instance Engine/Software* value and only if that too is not valid it will leverage the corresponding assumption.

Note: It is best that you provide the intended values for each database, since same defaults might not work the same for different engines.

For determining the RDS instances, the compute and memory specs are multiplied by the corresponding utilization method (On-Premises capacity, peak and average). Similar to servers, for peak and average compute and memory there are default utilization values that will be used whenever the input values are missing or regardless of the input values as per your settings in *Assumptions > AWS database recommendations*. As for servers analysis, you can specify which instance types to be avoided (in *Assumptions > AWS database recommendations > Do not recommend these RDS instance types*). You can review and override the RDS recommendations on the *AWS > RDS recommendations* page.

Note:

- If more databases are linked to the same server, MPA divides the specs of the server equally between those databases. This might not result in an optimal recommendation for all portfolios so be cautious of such cases.
- SQL Server BYOL is not supported on RDS, therefore MPA will not provide a recommendation for a

database with this target and license model. You can consider hosting those databases on DH, that means setting the tenancy as *Dedicated Host* for the corresponding server (see Dedicated Host recommendations).
- Likewise, Oracle Enterprise edition license included is not supported on RDS and MPA will not provide a recommendation for it. Consider changing to BYOL instead.

4.8.2. RDS cost estimation

When RDS recommendations is *On*, on the *AWS > RDS cost* page, MPA lists the total estimated RDS and license cost for the databases in scope. In the *Database cash flow* tab see the cash flow for the TCO duration for each database in scope for its recommended payment option. As for EC2, the residual is calculated for 3-year payment options if the TCO duration is not a multiple of 3.
Under the *Database pricing detail* tab, see the RDS cost for the TCO duration for each database in scope and each of the payment options selected in *Assumptions > AWS database cost*. Under *Database licensing*, see the estimated number of licenses required and the estimated license cost. For the cost, MPA uses the corresponding license cost, discount and support percentages for each target and edition as captured under *Assumptions > Database licenses*.
All costs are determined based on the instance type recommended given the utilization method selected under *Assumptions > AWS database cost > Database recommendation*.

Note:
- The license cost is not a cost to be paid to AWS, but rather to the vendor of the DB engine.
- Since the license cost is a function of the contract negotiated between the customer and the vendor, this cost can greatly vary from customer to customer. As all other MPA amounts, this too is provided as directional only.

Find the RDS storage cost under *AWS > Server & network cost*; there are separate line items for databases in summary and dedicated tabs in the *Details* section. MPA calculates database storage based on *DB Size* of the database and the total of the *Storage-Max Read/Write IOPS* of the corresponding server, or the corresponding fraction of it if the same server is shared by multiple databases. MPA first determines the type of volume required: if total max IOPS> 16000 (GP2 max size) or *DB Size* * 3 < total max IOPS then *Provisioned IOPS* is chosen, else *General Purpose* is recommended (*General Purpose-Aurora* if the target engine is *Aurora PostgreSQL* or *MySQL*).

Under *Database storage pricing* see for each database in scope, the recommended volume type, database storage size, server max IOPS and the corresponding annual cost. Under *RDS General Purpose, RDS General Purpose-Aurora* and *RDS Provisioned IOPS* see the annual cash flow for the duration of the TCO for the respective database storage.

Note:
- Double-check all databases for which *Provisioned IOPS* was recommended as this volume type is more expensive than *General Purpose* and data errors in the max Read/Write IOPS can lead to this.
- MPA is not yet considering the RDS snapshot cost or the Aurora I/O Rate cost.

4.8.3. Network cost estimation

MPA estimates the network cost *only* for the *Outbound Data Transfer* and presents it on the *AWS > Storage and network cost* page.
For each entry provided in *Assets > Network*, MPA uses the *AWS Egress Data Transfer(GB/Month)* if non-zero, else it uses the *Size of pipe (Mbits/second)* as follows: *Size of pipe (Mbits/second)* times *Outbound peak utilization* (in *Assumptions > AWS Network*) divided by *Peak/Average ratio* and converted to GB/month (conversion is: times 3600 seconds in a day times 30 days per month divided by 8 bits per Byte divided by 1024 MBs per GBs).
The cost for the outbound data transfer uses the tiered structure with the per unit cost decreasing as the usage increases. MPA applies the current tiers to compute the cost (find the public pricing on the S3 pricing website, the Data transfer tab).

Note: MPA doesn't estimate the intra-region or inbound network cost, only outbound data transfer.

4.9. On-Premises cost estimation

MPA estimates the On-Premises cost for compute, database, storage, network and IT labor, based on a logic and default assumptions aligned with the AWS Cloud Economics team.
By default, all the in-scope assets in your portfolio are assessed the On-Premises TCO. In *Assumptions* > *On-Premises server cost*, you can exclude servers with *Retire* migration strategy from the On-Premises TCO.
For customers that don't own their current infrastructure On-Premises, but rather in colocation, there can be cost differences. You can specify to use *Colocation* pricing in *Assumptions* > *General* > *Environment type*.

4.9.1. Compute

MPA presents the estimated cost for the On-Premises compute under *On-Premises* > *Server cost*. There is a summary section and detailed sections breaking down the costs between Physical and Host servers, rack infrastructure, software and license costs, facilities costs and a cash flow view. You will see here only those servers that are provided an EC2 Shared/Dedicated Instance or Dedicated Host. Servers provided an EC2 with SQL Server LI or RDS recommendation will be shown under *On-Premises* > *Database cost*.

MPA determines the server cost based on industry cost approximations for compute and RAM, by mapping the On-Premises specifications to a predefined set of configurations. For virtual servers, MPA first determines the number of physical hosts that are required and estimates their cost. You define the host types that MPA should consider, through *Configure host types* on the *Host servers* tab on either the *On-Premises* > *Server cost* or the *On-Premises* > *Database cost* pages (same host types are accessible from both pages).

If you haven't specified a host type through the *Server-Target Host Type* attribute at server level, MPA will pick the most cost-effective one: for each host type MPA determines the number of hosts that would be needed and then selects the host type with the lowest overall estimated cost. The calculation of the number of hosts needed takes the maximum between the number of hosts required to provide the compute needed vs the number of hosts required to provide the memory needed. The number of hosts to provide the compute needed is the total number of cores (processors times cores per processor) for all the VMs in scope, divided by the compute capacity of the host type which is its total number of cores times its number of threads times its CPU subscription ratio. Similar for memory. The cost for the hosts is calculated based on a table of price per compute configuration plus the estimated cost of the memory adjusted for the hardware discount (*Assumptions* > *On-Premises discount/adjustment* > *Server and hardware discount*) and accounting for region variability. The table and the memory price per GB, as well as the region variability are built-in (not configurable by the user) and are defined based on industry best practices.

4.9.2. Rack infrastructure

MPA estimates the cost of the rack infrastructure itself; this consists of the cost of racks, PDUs (Power Distribution Units), ToR (Top of Rack Switches), 1-time server deployment, spare capacity and the rack operations cost.

The PDU cost is calculated as twice the number of racks times the unit price per PDU as configured in *Assumptions* > *On-Premises rack infrastructure* > *Power distribution unit cost*, adjusted for the AWS Region based on built-in, per region, coefficients.
The ToR cost is computed as the number of racks times the number of switches per rack and the yearly switch support cost (in *Assumptions* > *On-Premises rack infrastructure* > *Number of ToR switches per rack* and *ToR switch support cost per year*).
The 1-time server deployment cost is computed as the total number of physical servers and physical hosts times the server deployment cost (in *Assumptions* > *On-Premises rack infrastructure* > *Server deployment cost*), adjusted for the AWS Region.
The spare capacity cost is computed as the spare capacity percentage (in *Assumptions* > *On-Premises rack infrastructure* > *Spare capacity for server per year*) applied to the total server cost plus maintenance.
The server maintenance cost (in *On-Premises* > *Server cost*, *Server maintenance contract* line item in summary) is computed as the hardware maintenance rate (*Assumptions* > *On-Premises rack infrastructure* > *Hardware maintenance rate per year*) applied to the total server cost.
The rack operations cost is estimated as number of racks times the configured monthly operation cost corresponding to the environment type (in Assumptions > On-Premises Rack Infrastructure > On-Premises/Colocation Monthly Cost to Operate a Rack).

4.9.3. OS and hypervisor licensing

The *Software and licenses* detail tab on the *On-Premises* > *Server cost* page presents the estimated costs for the OS and hypervisor licensing. MPA estimates Windows, SUSE, RHEL, Microsoft Hyper-V and VMware vSphere license costs. The license costs are computed by determining the number of licenses then multiplying with the corresponding license cost (in *Assumptions* > *Software and licenses*) and applying the software discount (in *Assumptions* > *On-Premises discount/adjustment* > *Software discount*). For Windows and Hyper-V, MPA also computes the yearly maintenance cost (Software Assurance) by multiplying the license cost with the *Windows 2016 Datacenter Software Assurance percentage* in *Assumptions* > *Software and licenses*. Note that all OS and hypervisor licensing costs are estimated separately for the servers running databases and those not running databases and shown on the corresponding *Software and licenses* pages.

For physical servers running Windows, the number of licenses is determined as the sum of number of on-prem CPUs times number of cores for all Windows physical servers; minimum 8 cores per CPU, minimum 16 cores per server. Note that the Windows license cost is per pack of 2 cores (in *Assumptions* > *Software and licenses*).
For physical servers running SUSE or RHEL, the number of licenses is the number of servers divided by 2, minimum 1. This is because the SUSE and RHEL licenses are priced per 2 CPUs.

For virtual servers, MPA first determines the number of physical hosts needed for the virtual machines (see above Compute), and then proceeds with the same license logic as for physical servers but applied to hosts.
MPA distinguishes between Windows guest OS licenses (*OS Name* + *OS Vesion* evaluates to *Windows* and *Hypervisor* is not *Hyper-V*) and Microsoft Hyper-V licenses (the *Hypervisor* has *Hyper-V* value, regardless of the OS). However, both are based on the same unit price and maintenance cost, those for *Windows 2016 Datacenter* in *Assumptions* > *Software and licenses*.

If the *Hypervisor* value evaluates to *VMware*, MPA estimates the VMware license cost. MPA supports both *vSphere (legacy)* solution which has one-time price per CPU and annual support and subscription cost, as well as solutions with only per-core annual subscription: *VMware vSphere Foundation, VMware Cloud Foundation* and *VMware vSphere Standard.* You specify the solution at portfolio level in *Assumptions* > *Software and licenses* > *Enable VMware Solution*, with the default being *vSphere (legacy).* Note that MPA evaluates the *Hypervisor* value as *VMware* when it is neither "HyperV", "Xen" or "KVM", case insensitive.
The VMware license cost is then calculated as: the cost in *Assumptions* > *Software and licenses* > *vSphere license cost* times the number of licenses (units) for *vSphere (legacy)*, and, for all other solutions, as the cost per corresponding VMware solution in *Assumptions* > *Software and licenses* times the number of licenses (units) times number of times the license duration assuming the license would be paid every 5 years. In both cases MPA applies the discount percentage specified in *Assumptions* > *On-Premises discount/adjustment* > *Software discount.* MPA computes the number of VMware licenses (units) as the number of CPUs of the corresponding hosts for *vSphere (legacy)*, and the number of cores for those hosts for all other solutions.
For all solutions except *vSphere (legacy)*, if the TCO duration is not a multiple of 5, MPA computes the residual amount for the number of years remaining until the next multiple of 5. For *vSphere (legacy)* only MPA also computes an annual support cost as the license cost times the percentage in *Assumptions* > *Software and licenses* > *vSphere support and subscription cost* times the TCO duration.

Example: *vSphere (legacy)*
Assume the following: one-time cost as set in assumptions is $3595, a support and subscription cost of 25%, the physical hosts hosting the VMware VMs have 56 CPUs, an on-prem discount of 25%, with TCO duration of 6 years. MPA will show *License cost* of 3595 * 56 * 0.75, a *Residual* of 0 and a *Total cost* as the difference between the two (150,990). In addition, the MPA shows the *VMware vSphere support Total cost* as 0.25 * 6 * 150,990.

Example: *vSphere Foundation*
Assume the following: annual cost as set in assumptions is $135, the physical hosts hosting the VMware VMs have 448 cores, an on-prem discount of 25%, and the TCO duration is 6 years.
MPA will show *License cost* of 135 * 448 * 10 * 0.75, a *Residual* of 135 * 448 * 4 * 0.75 and a *Total cost* as the difference between the two. Note that since the TCO duration is 6 years, the license duration is 10 years (paid in the 1st year and then again in the 6th, each time covering 5 years). Note also that MPA will not show the *VMware vSphere support Total cost* as it doesn't apply to this type of solution.

Note: If the VMs are more powerful than the hosts, then the calculation does result in more hosts being needed than the number of VMs. You can change those host configurations by clicking *Configure host types* button on the *Host servers* tab, on the *On-Premises > Server cost* page.

4.9.4. Database

MPA estimates the cost for the On-Premises infrastructure required to support a database only if that database is given an AWS recommendation; it presents the results on the *On-Premises > Database cost* page. This page follows the same structure as the *On-Premises > Server cost*, except that the information provided pertains to the database servers only. In addition, under the *Software and license* view, it also displays an additional section for the *Database engine software and licenses*. For the later, MPA uses the corresponding license cost, discount and support percentages for each target and edition as captured under *Assumptions > Database licenses*.

Note: On the *Host servers* tab, the *Configure host types* button leads to the same configurations as its corresponding peer under *On-Premises > Server cost*.

4.9.5. Database licensing estimation

A particular capability of MPA is that of computing database license costs, for both AWS and On-Premises. Since customers can have complex database licenses situations, by default MPA has this feature disabled. You can enable it in *Assumptions > Database licenses*. When enabled, MPA estimates the On-Premises database licenses only if an EC2 with SQL Server LI, EC2 shared/dedicated instance with Oracle BYOL, RDS or Dedicated Host recommendation is provided for that database (regardless of the `License Model` value).

Note: MPA only computes the database licenses in AWS when applicable. For EC2 with SQL Server license included, the license cost is included in the cost of the instance (see also Recommendations for EC2 with SQL Server license included). For RDS recommendations, only for Oracle bring your own license (BYOL) MPA estimates the 3rd party cost to run this in RDS; in this case, if Multi-AZ is specified for RDS target deployment, MPA is assuming replication is set up for On-Premises to keep the AWS and On-Premises costs comparable. For Dedicated Host, MPA estimates the 3rd party license cost only for SQL Server databases (Dedicated Host - 3rd party license estimation).

The On-Premises number of licenses is computed as the On-Premises vCPUs (from the server associated with the DB) times the Oracle Core Factor (in Assumptions > Database licenses). The in-RDS number of licenses is computed as the number of vCPUs of the recommended instance divided by a built-in factor of 2 for Enterprise and 4 for Standard edition.

For On-Premises license calculations, MPA assumes the customer is currently paying licenses, regardless of the license model provided in at database level or the corresponding assumption. The number of licenses is the number of vCPUs of the server, for physical servers, or host, for VMs, times (for Enterprise edition) the Oracle Enterprise Core Factor; the latter is determined based on the server's *CPU-Type* using the Oracle guidance for Enterprise Core Factor; if the CPU-type is not provided, MPA uses that value from *Assumptions > Database licenses > Oracle Enterprise Core Factor*.

Note:

- In its database assessment capabilities MPA is currently not supporting on-prem replication. There is one implementation aimed to reduce the on-prem vs AWS cost for Oracle BYOL licenses: when the target architecture requires Multi-AZ, and the target engine is *Oracle* with BYOL license option, the RDS license cost doubles; to keep the comparison between on-prem and AWS relevant, MPA is assuming, for this context only, that replication was used on-prem for those Oracle databases as well - keep this in mind when dealing with Oracle BYOL databases.
- The Multi-AZ is indicated by the *DB Instance Deployment Option* attribute of the database or the corresponding default value in *Assumptions > AWS database recommendation*.
- The license model is indicated by the *License Model* attribute of the database or the corresponding default value in *Assumptions > AWS database recommendation*.

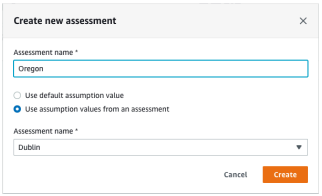
- SQL Server Express edition is free to use, however the Web edition cost is subject to vendor/customer agreement. If you know the customer is paying such cost and it is changing after the migration, then you can enter the different amounts as AWS and On-Premises adjustments (see Adjustments).

4.10. Network cost

MPA estimates the network cost as: the bandwidth cost, the network hardware and software cost, and the maintenance cost. Find these estimates on the *On-Premises* > *Storage and network cost* page.
The bandwidth cost is estimated differently depending on the on-premises *Environment type* as defined in *Assumptions* > *General*. When the environment type is *On-Premises*, MPA computes the yearly bandwidth for each entry in *Assets* > *Network* as: the *Size of pipe (Mbits/second)* times *On-Premises bandwidth price ($) per Mbps/month* (in *Assumptions* > *On-Premises network*) times *Multiplier for dark fiber to Internet exchange* (in *Assumptions* > *On-Premises network*) times 12 months per year. If the entry specifies that there is a redundant pipe, then this cost is doubled. When the environment type is *Colocation*, MPA uses a codified list of network configurations each with minimum transfer rate and corresponding price per Mbps/month and monthly recurring charge. For each entry in *Assets* > *Network* MPA identifies the configuration with the highest minimum transfer rate lower than the input *Size of pipe* and uses its prices to calculate the yearly costs as follows: a) bandwidth cost as *Size of pipe (Mbits/second)* times price per Mbps/month times 12 months per year and divided by the *Peak/Average ratio* of the entry, and b) the collocation recurring charges as the configuration's monthly recurring charge times 12 months per year.

The network hardware and software cost is calculated in the same way regardless of the on-premises environment type, as the *Network hardware and software cost (%)* (in *Assumptions* > *On-Premises network*) times the total costs of the hosts (sum of the cost of the physical and host servers as shown on the *On-Premises* > *Server cost* page).
Similarly, the network maintenance cost is the *Network hardware and software maintenance cost (%) (%)* (in *Assumptions* > *On-Premises network*) times the total costs of the hosts (sum of the cost of the physical and host servers as shown on the *On-Premises* > *Server cost* page).

4.11. Assessment comparison



You can create multiple TCO and Migration cost analyses, called assessments, on one portfolio without importing the data multiple times. Each assessment uses the same on-premises assets and dependencies, but have specific assumptions and server target architecture attributes.
To create a new assessment, on any TCO or Migration cost page click *Actions* > *Create new assessment*. You can leverage the default set of assumptions or inherit the values from a previously created assessment.
By default, each portfolio has an original assessment called *Assessment 1*. You can change its name and the name of any assessment in *Actions* > *Edit assessment*.
To compare the TCO results of all assessments, select *Actions* > *Compare assessments*.

4.12. Troubleshooting TCO results

4.12.1. EC2 recommendations

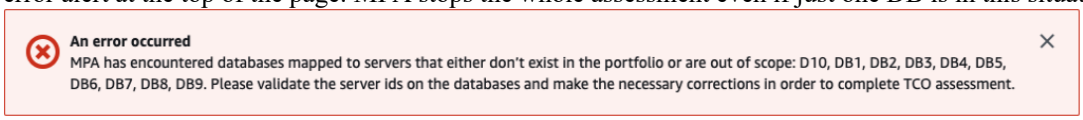
1. I am not seeing some of my servers showing up at all in the EC2 recommendations table (*AWS > EC2 recommendations, Details* section *EC2 instances* tab). Common reasons include:
   - Servers are marked as not in scope in *Assets > Servers*.
   - Servers have migration strategy *Retire* or *Retain*.
   - Servers are considered for *RDS recommendations* instead (*Enable RDS recommendations* is *On* in *Assumptions > AWS database recommendation*).

   For the first 3 reasons you will see details in the info box on the *AWS > EC2 recommendations* page.

2. Some of the servers are showing no recommendation at all for one or more of the utilization types. See common reasons for this in EC2 sizing.
3. The instance type that I've specified in *AWS - Instance Type* at server level is not coming up in recommendations. If the instance type is not valid, or not supported in that AWS Region, MPA silently defaults to using the AWS service to find out the proper recommendation.
4. I set my server's tenancy as Dedicated Host, but I don't see it in the servers with Dedicated Host recommendations. Dedicated Host has least priority between the MPA recommendation types, therefore:
   - Check if that server has a database in scope with target `SQL Server` and EC2 with SQL Server LI is enabled in *Assumptions > AWS database recommendations*; if it qualifies for EC2 with SQL Server LI you will see it along with the EC2 Shared/Dedicated Instance recommendations and costs.
   - Check if that server has a database in scope and RDS is enabled in *Assumptions > AWS database recommendations*; if an RDS recommendation is given, you will see it on the corresponding RDS pages.
5. EC2 instance cost comes as $0. Most likely this is due to the *Server-Usage* (server uptime) attribute in *Assets > Servers*. MPA multiplies this with the calculated On-Demand cost. If this attribute is set to 0, and given On-Demand is enabled by default, $0 would be lower than the estimated cost of any other payment option thus the final cost will come as $0 too. Note that by default, when *Server-Usage* is not populated, MPA assumes a server uptime percentage as per the value in A*ssumptions > AWS server cost > Default production/non-production server usage (uptime)*, where production/non-production is determined based on the *Environment Type* server attribute (or *Environment Name* if the type is missing).

4.12.2. RDS recommendations

1. I am not seeing any data in *AWS > RDS recommendations*.
   - Check that *Enable RDS recommendations* is turned *On* in *Assumptions -> AWS database recommendation*.
   - Check you have server ids specified for your databases (case sensitive) and that corresponding entries exist and are in scope under *Assets > Servers* with those server ids. When this issue happens, you will see an error alert at the top of the page: MPA stops the whole assessment even if just one DB is in this situation.

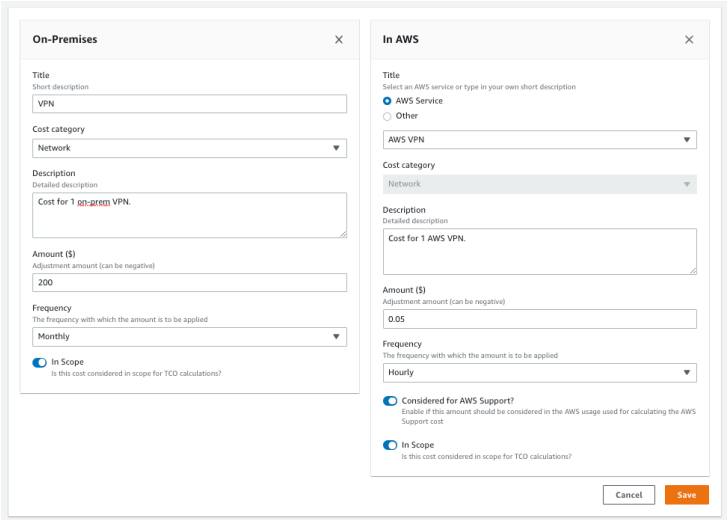
   - Check that your databases with *SQL Server* as target engine don't have *BYOL* as *License Model* when the edition is *Enterprise*.
   - Check that your databases with *Oracle* as target engine don't have *License included* as *License Model* when the edition is *Enterprise*.
   - Check that the *CPU-Number of Processors, CPU-Cores per Processor, CPU-Number of Threads per CPU-Core* and *RAM-Total Size* of the corresponding server, after accounting for utilization don't exceed the maximum supported values for RDS instances. For physical servers, MPA multiplies the total number of threads by 2. See here the list of RDS instance types.
2. I am getting a too low or even negative savings on AWS compared to on-prem. What can be the cause?
   - Check for servers that have high memory specification On-Premises compared to the number of CPUs, this can lead to RDS recommendation of instances that have a higher number of vCPUs than on-prem, hence the BYOL license on AWS gets to be more expensive than on-prem.

- Check for high database to server ratio, that is more than 3 databases with the same server id. This leads to a higher RDS instances being recommended, hence a higher cost.
- Check for Oracle BYOL: for *Enterprise* edition MPA is using the CPU type to determine a core processing factor which is used to compute the number of On-Premises licenses required, as per the Oracle guidelines. You can update it in *Assumptions* > *Database licenses* > *Oracle Enterprise Core Factor*.
- The BYOL cost is highly dependent on customer/vendor agreements. You can turn off assessment and inclusion of the license cost in TCO in *Assumptions* > *Database licenses* > *Include database license assessment in TCO*.
- The RDS cost, when comparing with similar EC2 instance type, is higher that the EC2 cost. For SQL Server and Oracle, consider migration to RDS Aurora or look for options of hosting on EC2.
- Check for DB storage (size): it should not be larger than the server size to which the DB is linked. MPA computes the On-Premises storage based on the server size, while the RDS storage is based on the DB size.

4.13. Adjustments

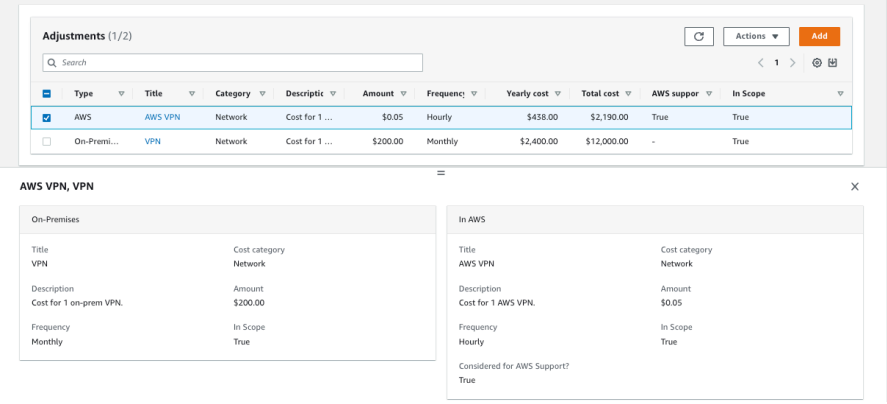
As MPA intendeds to provide a directional TCO, its focus is the most common and, usually, the biggest costs. Should you want to create a more in-depth business case, you can input additional costs to be include in the total TCO. Do this on the *TCO* > *Adjustments* page, click the *Add* button.
MPA allows you to enter either an On-Premises adjustment, an AWS one, or both. Entering both in the same time is useful when you are tracking corresponding costs between On-Premises and AWS.

For On-Premises, you specify the *Title* for the adjustment, should be a unique value across all adjustments in the assessment. MPA uses the *Cost category* to know where in the TCO summary to include this amount. Along with the amount, you specify the frequency with which it is applied. If later you need to exclude this adjustment from the TCO, you can update the *In scope* toggle.
For AWS, you can either select from a list of AWS services (in which case MPA sets the *Cost category* automatically) or specify your own title (you need to select the *Cost category* as well).
For AWS you also get to indicate if the amount should be considered for the AWS support – not all AWS costs are relevant for the AWS Support cost basis, see here for details.



Note that adjustments are assessment specific: in a portfolio with multiple assessments, each assessment can have its own adjustments and changes in one will not impact the other.

Once you save the adjustments, on the *TCO* > *Adjustments* page you can perform the usual activities: Delete, Edit, Download. When selecting just an adjustment record, MPA shows the On-Premises and AWS details side-by-side.



Note that MPA computes and presents in the table the yearly and total costs (for the TCO duration as defined in *Assumptions* > *General*). **However, just like with changes in *Assets*, you need to press *Calculate* for the adjustments to be considered in the TCO results!**
After *Calculate* you will find the adjustments included in the *TCO* > *Summary* in the corresponding cost categories.

4.14. TCO download

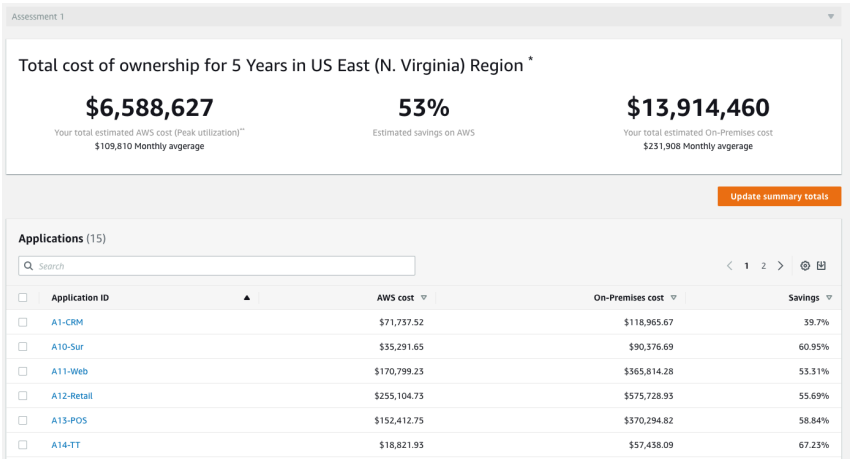
You can get an Excel file with the TCO details by clicking the *Download* button in the TCO banner on the TCO pages. The file includes four separate worksheets:

- The *Summary* represents the same information as on the *TCO* > *Summary* page with side-by-side total and cash flow comparisons, as well as details broken down between servers, databases and storage in AWS and On-Premises.
- The *AWS Details* worksheet lists all the servers in the portfolio, some of their On-Premises attributes, the application they are linked to, the database they are linked to, and the details of the AWS recommendation with focus on the EC2 and storage aspects. **Note that it doesn't include RDS details**.
- The *Financial Comparison* worksheet takes an opinionated approach to comparing stay On-Premises costs versus the AWS costs plus the migration costs. The summary table is prepared on an amortized basis. The transition from the On-Premise environment to AWS is based on quarterly percentages that impacts the *Parallel Run Costs* (On-Premise costs excluding infrastructure repurchases), *Migration Effort*, and *AWS Costs*. The amortized costs are estimates based on the data provided, industry benchmarks, and customer input (where applicable).
- The *Assumptions* worksheet lists all the assumptions in the assessment selected at the time the download was generated. It highlights in blue those values that are changed from the default MPA assumptions set, which can be useful when comparing results from two different portfolios with the same date.

Note: There is a separate download for the *Mainframe business case* from the banner on the *TCO* > *Mainframe summary* page.

5. Application cost

MPA estimates the cost of an application based on all the assets in the portfolio and their relations with the applications, as captured through dependencies. You can see a summary for the estimated cost of each application for the period of the TCO duration on the *TCO > Application cost summary* page. MPA provides the estimated cost for the entire TCO duration for each application in scope. If not all resources (servers and databases) are linked to all the in-scope applications, the cost of the remaining resources are allocated to a generic application called *Unmapped Resources*.



You can select any number of applications and click *Update summary totals* to see only the cost of those applications reflected in the *Total cost of ownership* banner. If the portfolio has more than one assessment (see Assessment comparison), you will be able to toggle between then from the dropdown above the banner.

Click the application's link to see the details for that particular application; this takes you to the *Application cost detail* page. Another way to get here is following the application's link in *Assets > Applications*.

5.1. AWS costs estimation

For estimating an application's costs, MPA takes a differentiated approach between the direct and common costs. The direct costs are costs associated with the application's resources which can be directly tied to those resources. *If a resource is shared among multiple applications, each application gets equal part of the resource's direct cost*. The common costs are not as easy to associate with specific application resources, therefore they are equally allocated to all applications that have direct costs.
MPA estimates the Compute, License and Storage direct costs and Labor, Network, Shared storage, Support and manually adjustment costs.

5.1.1. Direct costs

- Compute
  - The AWS Compute cost of a server is the EC2 cost of that server. This is also shown in *AWS > EC2 cost > EC2 pricing detail*: filter by the server id and look at the cost value corresponding to the *Recommended payment option* for that server. This also applies for dedicated host recommendations, where the EC2 cost of a server is computed as the total host cost times the ratio of vCPUs of the instance recommended for that server into the total number of used vCPUs of the host.
  - The AWS Compute cost of a database is the RDS cost of that database. This is also shown in *AWS > RDS cost > Database pricing detail*: filter by the database id and look at the cost value corresponding to the *Recommended payment option* for that database.
- OS License - 3rd party
  - The OS License - 3rd party is the BYOL cost for servers with *Windows* OS and *Dedicated Host* tenancy (shown as total for all hosts in *AWS > EC2 cost > Software and licenses*), distributed among the servers

sharing the same host based on the ratio of number of vCPUs of the instances recommended for each, into the total number of used vCPUs on the host.

- Database License - 3rd party
  - The AWS license cost is the cost of the database licenses the customer has to pay for BYOL licenses. This consists of RDS BYOL licenses (shown as total for all hosts in *AWS > RDS cost > Database licensing*: filter by the database id and look at the total cost) and Dedicated Host BYOL licenses (shown in *AWS > EC2 cost > Software and licenses*).
- Storage
  - The AWS Storage cost of a server is the storage cost of that server. This is also shown in *AWS > Storage & network cost > EBS volume per server*: filter by the server id and look at the total cost.
  - The AWS Storage cost of a database is the storage cost of that database. This is also shown in *AWS > Storage & network cost > Database storage pricing*: filter by the database id and take the annual cost times the TCO duration as defined in *Assumptions > General > Analysis duration*.
- RI Discount
  - The Reserved Instance Volume Discount distributed between those applications that have servers to which the recommended option is an RI volume discount eligible option, meaning it is not On-demand and not Convertible. The distribution is done proportionally with the server's RI amount into the total RI spent.

Note:

- Database costs (compute, license, storage) are currently only calculated if *Enable RDS recommendations* is *On* (in *Assumptions > AWS database recommendation*). This applies to both On-prem and AWS costs.

5.1.2. Common costs

- MPA uses the direct cost ratio, that is the ratio of the application's direct costs into the sum of all direct costs.
- The Labor cost is computed as the application's direct cost ratio applied to the AWS Labor cost as shown in *AWS > IT labor & support cost*.
- The Network cost is computed as the application's direct cost ratio applied to the AWS Network cost as shown in *AWS > Storage & network cost > Total AWS network cost*.
- The AWS Support cost is computed as the application's direct cost ratio applied to the AWS Support cost as shown in *AWS > IT labor & support cost*.
- The Manual adjustment is computed as the application's direct cost ratio applied to the AWS Manual adjustment as shown in *TCO > Summary > Manual adjustment*.
- The Shared storage cost is computed as the application's direct cost ratio applied to the shared storage cost as shown in *AWS > Storage & network cost > Object storage and Elastic File System* costs.

5.2. On-Premises costs estimation

MPA estimates the Compute, License, Software and Storage direct costs and Facilities, Hardware, Maintenance, Labor, Network, Shared storage (Backup, Rack and Software) and manually adjustment costs.

5.2.1. Direct costs

- Compute
  - The On-prem Compute cost of a physical server is the cost of that server as shown in *On-Premises > Server cost > Physical servers*: filter by the server id and look at the Server cost.
  - The On-prem Compute cost of a VM, is calculated from the total Host cost (in *On-Premises > Server cost > Host servers > Total cost*) for the server's host type, multiplied with the ratio of the determinant factor.
  - The On-prem Compute cost of a database hosted on a physical server is the cost of that server as shown in *On-Premises > Database cost > Physical servers*: filter by the server id and look at the Server cost.
  - The On-prem Compute cost of a VM, is calculated from the total Host cost (in *On-Premises > Server cost > Host servers > Total cost*) for the server's host type, multiplied with the ratio of the determinant factor.
- License
  - The license cost for servers comes under Software costs.

  - The On-Premises License cost of a database is the license cost of the database; MPA currently doesn't present this at database level, you can see the totals in the *On-Premises > Database cost > Software and license > Database engine software and license*).
- Software
  - The On-Premises Software cost of a server is the sum of the OS and hypervisor (if VM) costs of the server; MPA currently doesn't present this at server level, you can see the totals in the *On-Premises > Server cost > Software and license*.
  - For a VM, the OS and hypervisor costs are computed at host level and distributed to servers based on the determinant factor (vCPUs or RAM).
  - The On-prem Software cost of a database is the sum of the OS and hypervisor (if VM) costs of the corresponding server; MPA currently doesn't present this at database level, you can see the totals in the *On-Premises > Server cost > Software and license*).
- Storage
  - The On-prem Storage cost of a server is the storage cost as shown in *On-Premises > Storage & network cost > Server storage*: filter by the server id and look at the Cost value.
  - The On-prem Storage cost of a database is the storage cost as shown in *On-Premises > Storage & network cost > Database storage*: filter by the server id and look at the Cost value.

5.2.2. Common costs

- Facilities
  - The Facilities cost for servers is the Facilities costs for those servers that are not database servers as shown on *On-Premises > Server cost > Facilities*, adjusted by the application's server direct cost ratio, which is the ratio of the direct costs of the non-database servers of the applications into the direct costs of all non-database servers.
  - The Facilities cost for databases is the Facilities costs for those servers that are database servers as shown on *On-Premises > Database cost > Facilities*, adjusted by the application's database direct cost ratio, which is the ratio of the direct costs of the database servers of the applications into the direct costs of all database servers.
- Hardware
  - The Hardware cost for servers is the Rack infrastructure costs for those servers that are not database servers as shown on *On-Premises > Server cost > Rack infrastructure*, adjusted by the application's server direct cost ratio, which is the ratio of the direct costs of the non-database servers of the applications into the direct costs of all non-database servers.
  - The Hardware cost for databases is the Rack infrastructure costs for those servers that are database servers as shown on *On-Premises > Database cost > Rack infrastructure*, adjusted by the application's database direct cost ratio, which is the ratio of the direct costs of the database servers of the applications into the direct costs of all database servers.
- Maintenance
  - The Maintenance cost for servers is the Maintenance costs for those servers that are not database servers as shown on *On-Premises > Server cost > Server maintenance contract*, adjusted by the application's server direct cost ratio, which is the ratio of the direct costs of the non-database servers of the applications into the direct costs of all non-database servers.
  - The Maintenance cost for databases is the Maintenance costs for those servers that are database servers as shown on *On-Premises > Database cost > Server maintenance contract*, adjusted by the application's database direct cost ratio, which is the ratio of the direct costs of the database servers of the applications into the direct costs of all database servers.
- Labor
  - The Labor cost is as shown on *On-Premises > IT labor cost*, adjusted by the application's direct cost ratio.
- Network
  - The Network cost is as shown on *On-Premises > Storage & network cost > Total On-Premises network cost*, adjusted by the application's direct cost ratio.
- Storage
  - The Shared storage cost is the sum of all SAN, NAS, Object storage and corresponding backup costs, storage rack and software, adjusted by the application's direct cost ratio. MPA doesn't currently present the

rack and software storage costs broken down for Shared storage, but total and all other storage costs can be found on *On-Premises* > *Storage & network cost*.

- Manual adjustment
  - The Manual adjustment is as shown on *TCO* > *Summary* > *Manual adjustment*, adjusted by the application's direct cost ratio.

Note: The application's direct cost ratio is the ratio of the direct costs of the application into all the direct costs.

6. SQL Server optimization

As a first step towards supporting a more comprehensive Optimization and Licensing Agreement (OLA) functionality, MPA is providing a *standalone* capability to assess the potential license savings for SQL Server workloads when migrating to EC2. It is standalone because the results are not incorporated into the broader TCO. The *TCO > SQL Server optimization* assessment is done in the same time as all MPA assessments and you trigger it by clicking the *Calculate* button after making changes to assets or clicking the *Apply* button in *Modify assumptions*. The analysis builds upon EC2 recommendations, it picks up any server that has at least one SQL Server database (the *DB Instance Engine/Software* attribute) linked to it and is not considered for either RDS recommendations, if those are enabled.

There are two techniques MPA employs for license savings: from EC2 instances being recommended based on the On-Premises compute and memory needs adjusted for utilization, and from applying instance consolidation for those instances that require less than 4 licenses (there is a Microsoft requirement for minimum 4 licenses per VM, which applies to EC2 instances as well). The per license cost is picked up from *Assumptions > Database licenses > SQL Server license cost*.

On the *SQL Edition optimization* tab, MPA is only presenting additional information to help you navigate this OLA aspect. On the *SQL Server optimization* tab, MPA presents the edition level summary and server level details for the two optimization techniques.

In the *Servers* details table, the *On-Premises Min License Requirements* shows the number of licenses needed for the server on- premises based on the provisioned specifications. The *AWS Optimized Min License Requirements* shows the number of licenses needed for the same server to run in EC2 on an instance determined based on the utilization metric selected in *Assumptions > AWS server cost > Server recommendation*. The *AWS Optimized Min License Requirements With Consolidation* shows the number of licenses after consolidation is applied at instance level. Currently, the consolidation only looks for those servers that otherwise would run on less than 4 vCPUs in EC2 and combining them to save on the minimum number of licenses.
In *Summary*, the *Optimized Count* represents the number of licenses after both techniques are applied.

When determining the On-Premises licenses needed, MPA distinguishes between Physical and Virtual servers in that, for Virtual servers, you can specify if they should be considered licenses at Host level (default) or at VM level; change this in *Assumptions > AWS optimization and licensing*. For *Host level* licensing, MPA uses as host types the ones defined on the *On-Premises > Server cost* page, the *Host servers* tab; configure the host types as relevant for your customer. For *Host level* licensing, MPA is not presenting the *On-Premises Min License Requirements* since it is not relevant at server level it being determined at physical host level.

Troubleshooting: my server is not coming up in the OLA assessment.

- Check to see that the server has at least one SQL Server database (the *DB Instance Engine/Software* attribute).
- Check to see if RDS recommendations are turned *On* and then if the intended server got picked up by either of them (in *Assumptions > AWS database recommendations* and *Assumptions > General* respectively).
- Check to see if it is marked as *Retire* or *Retain* (on the *Migration strategy > Recommendations* page).
- Check to see if the server has only replica databases on it - those are listed in the details grid, but not considered for savings.

7. Mainframe workloads assessments

MPA can provide a rough order of magnitude assessment of Mainframe workloads. Similar to the regular TCO, MPA starts with the entered data, uses default values set in Assumptions and generates TCO assessment, cash flow and migration cost estimation – all presented on the *TCO > Mainframe summary* page; a download can be generated from the same page, similar to regular TCO.

The data required needs to be entered manually in *Assets > Mainframes*. You can enter the data for the whole mainframe site, or only for specific workloads.
There are Assumptions groups specific for Mainframe logic, but MPA uses some of its other assumptions as well. For details on what is used and how, see the following sections.

Note:
- As for the rest of MPA, Assumptions are not just static values MPA assumes to be true, but rather values for which MPA has defaults, and you need to review and update as per the circumstances of each customer.
- In the same portfolio, you can have mainframe data as well as non-mainframe (also referred to as regular or x86) data in MPA and generate TCO for both. However, currently, Mainframe TCO results are not integrated with the rest of MPA; for this reason, all the pages under TCO, except for the *TCO > Mainframe summary*, don't contain any mainframe relevant information; similarly, Mainframe data is not considered in Migration strategy, Cost estimation or Planning.

Note!
- *The migration cost is a rough order of magnitude estimate. For a formal migration quote, contact AWS Professional Services or a Global System Integrator*. Similarly, the On-Premises and AWS values are estimates and limited by the data provided and the logic MPA considers. You can always enter additional costs (positive or negative amounts) in *Assumptions > Migration tools and schedule > Additional migration effort costs*.

You can see the AWS cost details on the Details tab in *TCO > Mainframe summary*. MPA presents the costs for each cost item it currently prices at mainframe workload and environment level; the *Yes/No* values in the *Disaster Recovery?* column indicate if the cost applies to regular or DR cost item.

The *Cash flow* tab on the *TCO > Mainframe summary* page shows a comparison between the On-Premises *no change* scenario and a *gradual migration* to AWS, assuming an equal proportion of the On-Premises workloads migrating to AWS each year.
Example:
- For a 3 years migration period (in *Assumptions > Migration tools and schedule > Project duration*) MPA assumes in the first year 1/3 of the workloads are migrated, in the second year 2/3 of the workloads, with all being migrated by the end of the 3rd year.

Note: The On-Premises cost goes to zero only after the migration is complete, as it is assumed that the mainframe workloads will be decommissioned entirely, no gradual reduction.

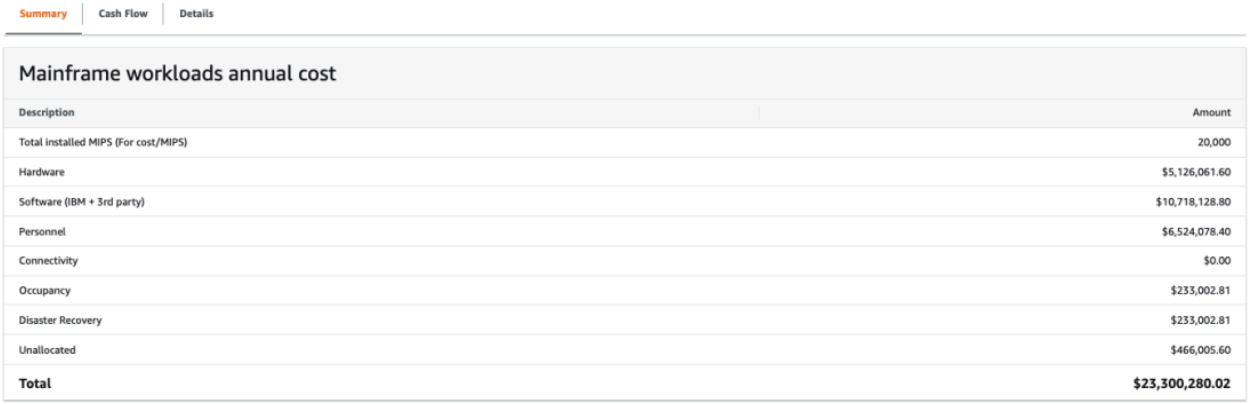
The tab also shows *Total annual costs* and *Cumulative costs* and the savings between the two scenarios.

7.1. Mainframe workloads annual cost

MPA determines the On-Premises cost for the total MIPS based on the per MIPS cost (in *Assumptions > Mainframe - On-Premises,* the *Small/Medium/Large MIPS average annual mainframe cost per installed MIPS*); the correct range is identified based on the *Total Installed MIPS* on *All* mainframe workloads, as specified in *Assumptions > Mainframe - On-Premises*.
The identified, per MIPS cost is multiplied with the total number of MIPS in all mainframe workloads, all environments. If you have entered a manual adjustment value in *Assumptions > Mainframe - On-Premises > Manual adjustment for the Mainframe On-Premises cost*, MPA will add that here. The combined cost of the MIPS and manual adjustment is then distributed between the different cost items (Hardware, Software, Personnel,

Connectivity, Occupancy, Disaster Recovery, Unallocated) using the corresponding percentages defined in *Assumptions > Mainframe - On-Premises*.

Note: MPA uses 2 concepts for installed MIPS: the *Total Installed MIPS* represents the maximum MIPS capacity of the machines, and is specified, for all mainframe sites for which data is entered, in *Assumptions > Mainframe - On-Premises*. The MIPS corresponding to the mainframe workloads for which data is entered; these values are as entered at workload and environment level in *Assets > Mainframes*.



Summary | Cash Flow | Details

Mainframe workloads annual cost

Description	Amount
Total installed MIPS (For cost/MIPS)	20,000
Hardware	$5,126,061.60
Software (IBM + 3rd party)	$10,718,128.80
Personnel	$6,524,078.40
Connectivity	$0.00
Occupancy	$233,002.81
Disaster Recovery	$233,002.81
Unallocated	$466,005.60
Total	**$23,300,280.02**

Note: The Connectivity cost is 0 because, in this example, the corresponding allocation percentage (in *Assumptions > Mainframe - On-Premises > Mainframe cost distribution - Connectivity*) is 0%.

7.2. Migration cost

Note!

- *The migration cost is a rough order of magnitude estimate. For a formal migration quote, contact AWS Professional Services or a Global System Integrator*. Similarly, the On-Premises and AWS values are estimates and limited by the data provided and the logic MPA considers. You can always enter additional costs (positive or negative amounts) in *Assumptions > Mainframe - migration costs > Manual adjustment for the Mainframe migration cost*.

The migration cost consists of the *SI or ProServe labor* costs and the *Tool license & services cost*.
The *Tool license & services cost* is computed as the *Total Number of Lines of Code* on *All* workloads (in *Assumptions > Mainframe - Migration costs*), times the cost per line of code based on the number of lines of code bracket (in *Assumptions > Mainframe - Migration costs > Tool cost per line of code - Smaller/ Small/ Medium ...*).

The SI or ProServe labor costs consist of:

- The labor migration support cost, computed as *Project duration* in months (in *Assumptions > Migration tools and schedule*), times the *Migration support cost per month* (in *Assumptions > Mainframe - Migration costs*).
- The labor cost for the integrated dependencies, computed as *Total integrated dependencies per environment* (in *Assumptions > Mainframe - Migration costs*), times total number of environments across *All* mainframe workloads, times the *Labor cost per integrated dependency per month* (in *Assumptions > Mainframe - Migration costs*).
- The cost of testing, computed as number of test cases times that cost per test case based on the number of tests bracket (in *Assumptions > Mainframe - Migration costs > Labor cost per test case - Smaller/ Small/ Medium ...*)

Note: The total number of environments refers to the sum of the max between *Application Environment Count* and *Database Environment Count* for the production environment of each workload as defined in *Assets > Mainframes*.

7.3. AWS annual cost

In all assessments MPA uses the *AWS Region* as specified in *Assumptions* > *General* > *AWS Region* and, for the total AWS cost that you can see in the banner summary on the *TCO* > *Mainframes summary* page, the TCO duration from *Assumptions* > *General* > *Analysis duration*.

MPA currently estimates the following AWS costs:

EC2 costs
MPA first estimates the number of EC2 instances that will be required: MPA uses the instance type (per environment type and workload type) specified in *Assumptions* > *Mainframe – AWS target architecture* > *<env> EC2 instance for <workload type>*. It then computes the number of instances required, per environment type and workload type (all assumption values are from *Mainframe - AWS target architecture*) as the max between the below result (rounded up) and twice the number of *Application Environment Count* (in input).

> *Average MIPS* (in input), times the *Online/Batch MIPS Share to Compute* (%),
> divided by the *Average Online/Batch Target CPU Utilization* (%),
> divided by the *MIPS to EC2 vCPU Ratio*,
> divided by the number of vCPUs of the instance type specified.

To get the EC2 estimated cost, for each environment type and workload type, MPA computes the *Unit monthly cost* as the lowest annual cost among all the payment options selected in *Assumptions* > *AWS server cost* > *EC2 payment types*, divided by 12; for the annual cost, MPA multiplies the hourly rate (where applicable) by 24 hours a day and 365 days a year.

Note: For *OnDemand* payment option, MPA further prorates the cost by the number of hours per week (*OLTP/Batch Hours*), considering 168 hours per week.

From all the EC2 offers of the selected payment types, MPA filters those that have the instance type specified in *Assumptions* > *Mainframe - AWS target architecture* > *<env> EC2 Instance for <workload type>,* the OS from *Assumptions* > *Mainframe - AWS target architecture* > *EC2 Instance OS*, and the tenancy from *Assumptions* > *Mainframe - AWS target architecture* > *EC2 tenancy*.

EBS cost
For each environment type, MPA assumes 1 EBS volume for each EC2 instance.
The average size of an EBS volume is defined in *Assumptions* > *Mainframe - AWS target architecture* > *Average EBS volume size*.
To get the EBS estimated cost, for each environment type, MPA computes the *Unit monthly cost* as the lowest monthly rate between all the volume types from *Assumptions* > *AWS storage* > *EBS volume types*, times the total size of the EBS volumes in GBs. The size of an EBS volume is as defined in *Assumptions* > *Mainframe - AWS Target Architecture* > *Average EBS volume size*.
MPA assumes 1 EBS volume is needed for each EC2 instance, and it assumes they are root volumes, hence ST1 and SC1 are not supported as volume types. MPA updates the EBS volume size to account for backup by considering the monthly backup/change rate from *Assumptions* > *AWS storage* > *Rate of change per month*, monthly growth rate from *Assumptions* > *General storage* > *Growth rate per month*, and the snapshot compression ratio from *Assumptions* > *AWS storage* > *EBS snapshot compression ratio*. The EBS snapshot compression ratio should be set based on the value the customer expects to get by implementing compression on the EBS snapshot files in S3.

RDS cost
MPA uses the instance type (per environment type) specified in *Assumptions* > *Mainframe - AWS target architecture* > *<env> RDS database instance*. It then computes the number of instances required, per environment type (all assumption values are from *Assumptions* > *Mainframe - AWS target architecture*) as the max between the below result (rounded up) and 2x the number of *Database Environment Count* (in input).

> [*Average Database MIPS* (input) + *Average Online MIPS* (input), times the *Online MIPS Share to Database* (%) + *Average Batch MIPS* (input), *times the Batch MIPS Share to Database* (%)] divided by the *Average Online Target CPU Utilization* (%), divided by the *MIPS to EC2 vCPU Ratio*, divided by the number of vCPUs of the instance type specified in <env> RDS database instance.

To get the RDS estimated cost, for each environment type, MPA then computes the Unit monthly cost as the lowest annual cost among all the payment options selected in *Assumptions > AWS database cost > RDS payment types*, divided by 12; for the annual cost, MPA multiplies the hourly rate (where applicable) by 24 hours a day and 365 days a year.
From all the RDS offers of the selected payment types, MPA filters those that have the instance type specified in *Assumptions > Mainframe - AWS target architecture > <env> database instance*, the DB engine from *Assumptions >AWS database recommendations > Default target database engine*, the DB edition from *Assumptions >AWS database recommendations > Database edition*, DB license model from *Assumptions > AWS database recommendations > Database license model*, and the deployment option from *Assumptions > AWS database recommendations > Default deployment option*.

RDS Storage cost
MPA determines the volume type for each environment, as follows:

- If any of the RDS Aurora options were selected in *Assumptions > AWS database recommendation > Default target database engine*, then the volume type is *General Purpose – Aurora*.
- If the Max IOPS > 16000 (the Max IOPS for GP2) or the Max IOPS > DB Size * 3, then MPA recommends Provisioned IOPS (IO1), else MPA recommends General Purpose (GP2). The Max IOPS is calculated as: Sum of the *Database Disk Used* (TB, from input) + *Files Disk Used* (TB, from input), converted to KBs, multiplied by a coefficient of 80%, and divided by 30*24*60*60 to get the Max IOPS.

The size of the DB is *Database Disk Used* (TB), from input.

To get the RDS Storage cost estimate, for each environment type, MPA computes the *Unit monthly cost* as the per GB rate for the region for the recommended volume type, times the number of GBs converted from the *Database Disk Used (TB)* in input.
For Provisioned IOPS volume type, MPA also computes the IOPS cost by multiplying the Max IOPS with the Provisioned IOPS rate for the region - this is only for non-Aurora engines, as MPA doesn't yet compute IO Rate for Aurora.

Object storage pricing
For each environment type, MPA uses:

- The number of S3 IA PUTs in *Assumptions > Mainframe - AWS target architecture > PUTs per TB of S3 IA*
- The number of S3 IA GETs in *Assumptions > Mainframe - AWS target architecture > GETs/LISTs per TB of S3 IA.*
- Monthly growth rate in *Assumptions > General storage > Growth rate per month.*

The *Unit monthly cost* for S3 Standard Storage is calculated as per the region's tiered rates applied to the *Files Disk Used (TB)* from input (converted to GBs).
The *Unit monthly cost* for S3 Standard IA is calculated as per the region's S3 IA rate, times the *actual* storage for the *HSM and SMS Tape (TB)* from input (converted to GBs); the actual storage reflects the S3 IA minimum object size requirement of 128 KB per object, and is calculated based on built-in assumptions: average size of an object is assumed 1MB (this allows MPA to estimate the total number of objects in IA), 10% of all objects are assumed to be smaller than 128 KB, which an average size per object of 20 KB. Because of the actual storage logic, the size considered for cost purposes is slightly higher (1.055% more) than the original *HSM and SMS Tape* (TB) from input.
The *Unit monthly cost* for Glacier is calculated as per the region's Glacier storage rate, times the *Backup Tape (TB)* from input (converted to GBs).
In S3 Retrieval MPA considers the costs for S3 IA PUTs, GETs and Retrieval of data from S3 Standard IA.
Note that MPA doesn't support at the moment Glacier retrieval.

Data transfer cost
For each environment type, MPA computes the *Data Transfer OUT From Amazon EC2 To Internet* (see also public pricing here) based on the *Data Transfer Out* (GB) from input per the corresponding tiers.

IT labor cost
MPA computes the IT labor pricing aggregated for all environments. It is the sum of:

- EC2 labor cost, calculated as the total number of EC2 instances, divided by the number of EC2 instances per admin (in *Assumptions > AWS IT labor & support > Number of EC2 instances per admin*), times the admin salary from *Assumptions > AWS IT labor & support > Administrator's annual salary*.
- Similarly, the RDS labor cost is calculated as the total number of RDS instances, divided by the number of RDS instances per admin (*Assumptions > AWS IT labor & support > Number of RDS instances per admin*), times the admin salary from *Assumptions > AWS IT labor & support > Administrator's annual salary*.
- The storage labor cost is calculated as the total number of TBs (object storage and EBS volumes), divided by the number of TBs per admin (*Assumptions > AWS IT labor & support > Storage size (TB) per admin*), times the admin salary from *Assumptions > AWS IT labor & support > Administrator's annual salary*.

AWS support cost
MPA computes the AWS support pricing aggregated for all environments. It is calculated using the corresponding tier % for the monthly average spend; tier values are based on the support plan specified in *Assumptions > AWS IT labor & support > AWS Support*.

Disaster recovery costs
MPA logic for Disaster Recovery is based on the disaster recovery mode specified in *Assumptions > Mainframe - AWS target architecture > Disaster recovery strategy*, and applies only for *Production* workload environments. If the value is *None*, no disaster recovery cost is estimated. For *Pilot Light*, MPA estimates only the costs for RDS and Object storage, while for *Warm Standby*, MPA estimates also the cost for EC2 and EBS using the same usage values as per the corresponding environment, except for EC2 where the number of instances assumed per workload is the minimum between the number of instances of the workload environment and the maximum between the *Application Environment Count* and *Database Environment Count* and the number *2*. Glacier storage costs are assessed for all DR types based on the *Backup Tape* value from input.

8. Migration strategy

8.1. Servers

MPA recommends the R-types for server migration strategy. Find the results in *Migration strategy* > *Recommendations*.

MPA logic is based on the server's OS *(OS Name + OS Version)*:

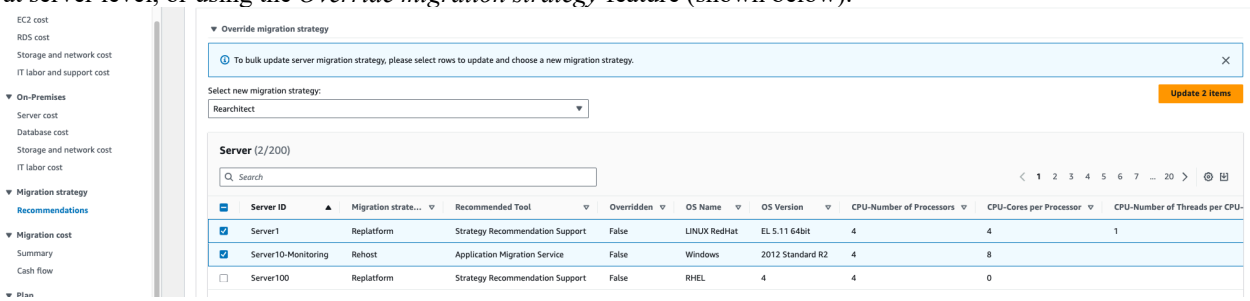
- Retire: if the OS contains ESX* (case insensitive)
  - Rehost: if the OS is supported by the AWS Application Migration Service and a value is not specified or is same as the on-prem value for the *AWS – OS* server attribute.
    - Replatform: if the server a) doesn't qualify for Rehost and the OS evaluates to Windows, RedHat, SUSE, Ubuntu, Debian, CentOS, FreeBSD, GNU, Linux OR b) *AWS - OS* is specified, and the OS is different than the *AWS - OS*.
      - Rearchitect: When none of the above apply.

* This is because the ESX OS is used for VMware host machines that will no longer be needed on-prem after migration.

Note:

- Different assessments in the same portfolio might have different migration strategy recommendations; they are assessment level, similar to how assumptions are assessment level.
- Relocate (VMware on-prem to VMware Cloud in AWS): As of April 30, 2024, VMware Cloud on AWS is no longer resold by AWS or its channel partners. The service will continue to be available through Broadcom. MPA doesn't assess VMware Cloud in AWS options anymore.

You can set your own values for the server R-type either through the *Server-Migration Pattern (Imported)* attribute at server level, or using the *Override migration strategy* feature (shown below).



8.2. Applications

Unless provided in input (*Assets* > *Applications*, the *Migration Strategy* attribute), the application migration strategy is determined based on the underlying servers: the application strategy has the value of the highest ranked server strategy among all servers mapped to that application. The strategies are ranked in this order: Retire, Rehost, Replatform, Rearchitect.

Example:

- An application is mapped to two servers, one having the migration strategy *Rehost* and the other one *Replatform*; the application strategy will be *Replatform*.

Note: If you get a value of *Unknown* for the application migration strategy, check if the application is mapped to a server. You can see this in the Application data page (the *Mapped* field) or in the *Dependencies* > *Server to application*.

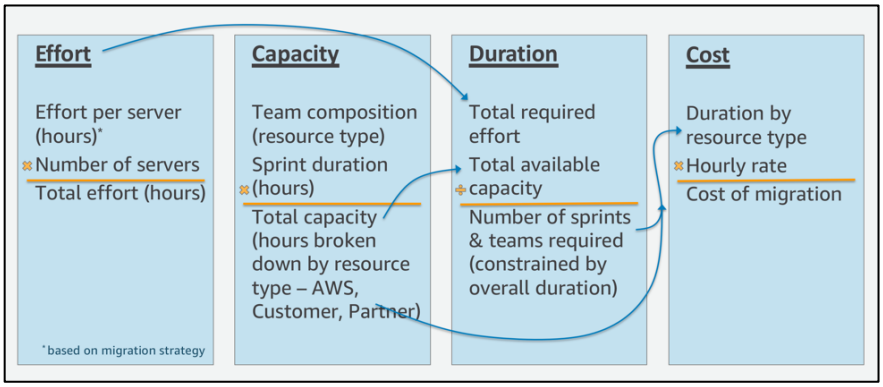
If the *Migration Strategy* attribute is provided in input, it is used as-is and you will see the *Overridden* attribute as *Yes* in *Migration Strategy* > *Recommendations*, the *Application migration strategy* tab.

9. Migration cost

9.1. Estimate

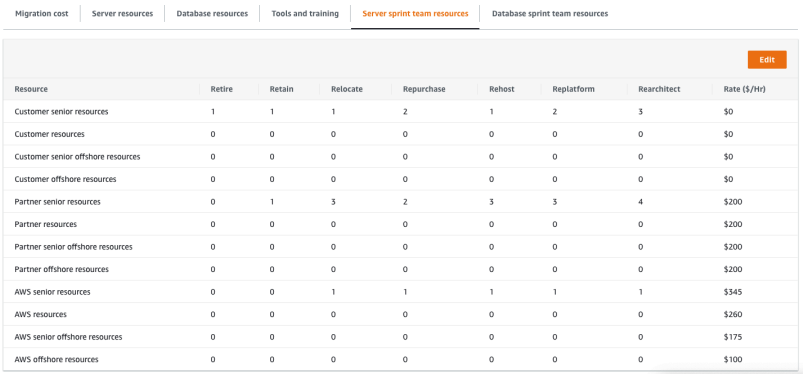
The *Migration cost* of executing an On-Premises to AWS migration project consists of the project management cost to which you can add the cost of migration and discovery tools, training and additional costs for a more complete picture.

MPA estimates the project management cost by considering the effort per R-type for your servers and databases, the project duration and available resources. The following model is used for the estimation at server and database level:



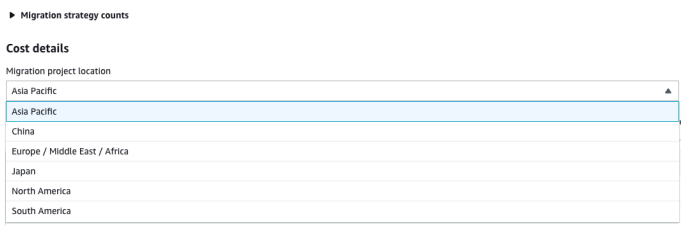
The project management cost for a resource type (*Customer/AWS/Partner*) is computed based on the number of hours of management per week as defined in *Assumptions* > *Migration tools and schedule* > *Management working hours per week* times the number of hours of the management sprints (max number of sprints for that resource type times the number of weeks per sprint in *Assumptions* > *Migration tools and schedule* > *Sprint duration*), all at a the cost rate of the senior resource (example: for partners, that is the *Partner senior resource* as defined on *Migration cost* > *Summary* > *Server/Database sprint team resources*). Since the management effort covers all the teams of the same resource, the above is distributed between the different R-types based on the ratio of the resources of that type (example: *Partner*) needed for the R-type among all the resources of that type needed for the migration, rounded down to the nearest thousand.

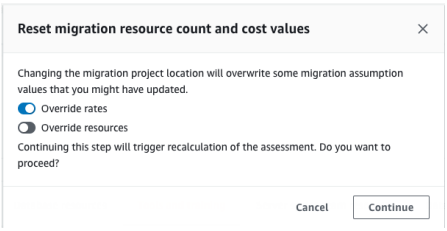
Find the results on the *Migration cost* > *Summary* page. Note that this page has a different banner than the TCO pages, it is specific to the migration cost and duration.
You can change the per R-type migration effort in *Assumptions* > *Server/Database migration level of effort*.
You can change the hourly rate and sprint team resource assignment by clicking the *Edit* button on the *Sever* and *Database sprint team resources* tabs.



Resource	Retire	Retain	Relocate	Repurchase	Rehost	Replatform	Rearchitect	Rate ($/Hr)
Customer senior resources	1	1	1	2	1	2	3	$0
Customer resources	0	0	0	0	0	0	0	$0
Customer senior offshore resources	0	0	0	0	0	0	0	$0
Customer offshore resources	0	0	0	0	0	0	0	$0
Partner senior resources	0	1	3	2	3	3	4	$200
Partner resources	0	0	0	0	0	0	0	$200
Partner senior offshore resources	0	0	0	0	0	0	0	$200
Partner offshore resources	0	0	0	0	0	0	0	$200
AWS senior resources	0	0	1	1	1	1	1	$345
AWS resources	0	0	0	0	0	0	0	$260
AWS senior offshore resources	0	0	0	0	0	0	0	$175
AWS offshore resources	0	0	0	0	0	0	0	$100

By default, MPA uses the servers and databases in-scope of your portfolio, but you can run the migration cost estimation without complete server/database data, indeed without any server and database records. Just enter the server and database counts directly in the UI broken down by each R-type.
Select the migration project location based on the location where the customer wants to run the migration project. MPA has default resource cost for each location. You will be prompted to choose whether to override the already entered resources and/or rates when changing the location.





To change the cost for the migration and discovery tools, training and additional costs see *Assumptions* > *Migration tools and schedule*.

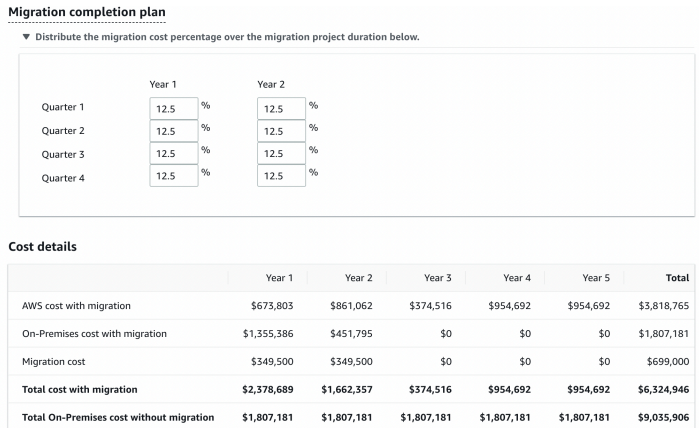
You can download the summary of *Migration cost* analysis by clicking the *Download* button in the banner. The file has the same data as seen in MPA itself: the migration cost summary by R-type, the server and database sprint team composition, and detailed server and database resource allocations.

9.2. Cash flow

The migration cash flow distributes the migration cost over the TCO duration (as defined in *Assumptions* > *General* > *Analysis duration*). Based on the projected quarterly percentage of migrated workloads, it estimates:

- The yearly On-Premises downsized cost (*On-Premises cost with migration*).
- The yearly AWS incrementally growing cost (*AWS cost with migration*).
- The yearly cost of the migration project (*Migration cost*).

As a simplifying approach, MPA applies the quarterly percentages to the total costs for the TCO duration, hence it assumes that costs are split evenly between all the workloads, hence what is actually migrated in a quarter doesn't make a difference, only how much is migrated matters.



For the *On-Premises cost with migration*, for each quarter, MPA considers that the cost is that corresponding to the total On-Premises cost without migration (as seen for example in the TCO banner) divided by the number of quarters in the TCO duration, reduced by the percentage of what was already migrated. The percentage of what was already migrated is computed as the quota allocated to all previous quarters, plus half of the quota for the current quarter; this is because workloads will get migrated throughout the quarter, hence, while all the quota would have

been migrated completely at the end of that quarter, costs are still incurred in various degrees throughout the quarter; MPA assumes that degree to be 50%, hence the half quota.
For example, with the default settings, in the 3rd quarter, 25% was completely migrated in the previous quarters, and 7.25% from current quarter, to a total of 32.25%.

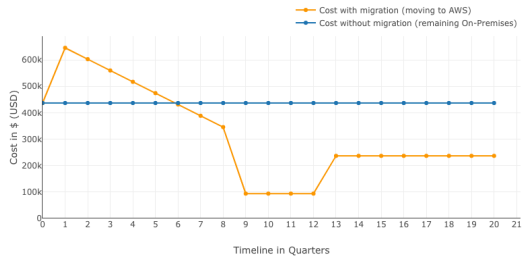
For the *AWS cost with migration,* MPA takes a more nuanced approach by treating separately the Reserved Instance upfront costs vs the monthly costs. The monthly costs follow a similar approach to On-Premises in that, for each quarter, MPA computes the monthly cost corresponding to what was already migrated plus half the quota of the current quarter. For upfront costs, MPA assumes they are fully paid at the beginning of the quarter for all that will be migrated in that quarter, and keeps track of the recurrence of what was already migrated. The recurrence is reflected through the number of times the cost is paid (first time and renewals) in the duration and when it gets renewed.

Example:

- If the total 1-year RI upfront costs for the TCO duration are $44,075 (5-year TCO), and the first quarter has a 20% quota, then the upfront cost for the first quarter is $44,075 / 5 * 20%; in the 5th quarter, assuming that has 10% quota, the cost will be a new $44,075 / 5 * 10%, plus the first quarter's renewal of $44,075 / 5 * 20%. The "/ 5" is due to the fact that 1-year upfront costs for a migrated payload will be paid yearly, hence 5 times in a 5-year TCO.
- In the same example as above, but assuming 3-year RI instead, if amount of $35,542, for the first quarter MPA computes the upfront cost as $35,542 / 2 * 20%, while for the 5th quarter it will be just $35,542 / 2 * 10% - there is no renewal of what was migrated in the 1st quarter yet, that will only happen in the 1st quarter of the 4th year of the TCO duration. The "/ 2" is due to the fact that 3-year upfront costs for a migrated payload will be paid only 2 times in a 5-year TCO, in the 1st year they are migrated and then again 3 years later.

The cash flow graph presents a quarterly view of the costs, helping highlight when the migration project starts yielding positive results, that is when the costs with migration become less than those without migration. In this example, the migration starts bringing savings already in the 7th quarter. Note that the cost with migration picks up again in the 13th quarter, that is because 3-year RI renewals kick in.

Estimated migration project cash flow



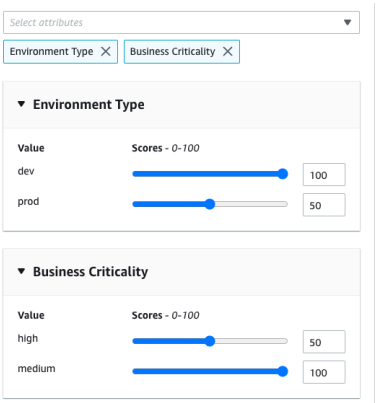
10. Application prioritization

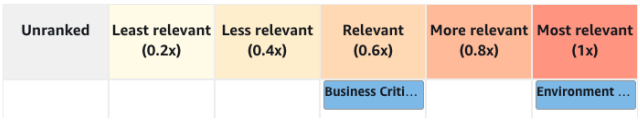
Typically, for large application portfolios, you need to rank the applications to identify the likely candidates for pilot migration, for example as part of the Mobilize phase of the AWS Migration Framework. MPA takes a score-based approach to application ranking and puts you in the driver's seat as the one defining the criteria.

- Decide which are the relevant application attributes for prioritization.
- Create new one if one doesn't already exist see Additional attributes)
- Collect the data for those attributes if not already available.
- On the *Plan* > *Application prioritization* page
    - Select and score attribute values.
    - Rank the attributes by relative relevance.
    - Review the results and edit the prioritization rules as needed.

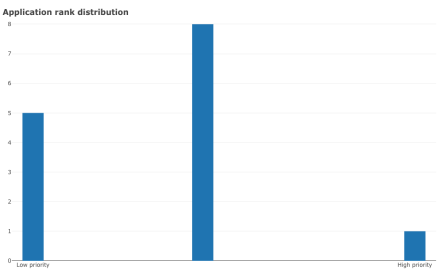
Example:

- Let's assume the customer wants to prioritize for pilot migration those applications that are not business critical and specifically their non-production environments. We select the *Business Criticality* and *Environment Type* attributes; we assign highest score to the *medium* criticality and lower score to the *high* criticality; similarly, we assign the highest score to the *dev* environment and lower to the *prod* environment.
- Let's also assume that the environment type is the deciding factor, that is: if an application is of high criticality, but in a dev environment, we might still want to consider that for pilot. For this reason, in the next step we set the *Environment Type* as the *Most relevant*, and the *Business Criticality* as just *Relevant*.





- Click *Prioritize applications*, and MPA ranks all applications based on these criteria.



The *Application rank distribution* graph presents the number of applications with each rank, with *High priority* being rank 1 and *Low priority* being rank 3.

MPA computes the rank of each application based on the sum of the weighted scores of the application's values for the attributes considered in prioritization. In the above example, an application with a *dev* environment and *high* business criticality will have a score of 100 x 0.6 + 50 x 1 = 110.
In the above example, there are 4 distinct total application scores possible: 110 (*dev/high*), 80 (*prod/high*), 160 (*dev/medium*) and 130 (*prod/medium*). There are only 3 distinct ranking groups (bars in the rank distribution graph) because one of the combinations didn't occur (no applications matched its values).
An application's rank is the order of its total score in all the distinct score values, with the highest score getting the 1st rank and so on.

Note: If more than 10 ranks are identified, in the rank distribution graph, MPA presents the applications grouped in maximum 10 groups based on a score normalization logic – this is just to simplify the presentation, the rank itself is not affected.

11. Dependency groups

Sometimes customers look to migrate everything they have, lift-and-shift. Often, they actually want to move specific applications in concentrated efforts referred to as *waves*. When moving applications in a wave, customers need to pay attention to the relations those applications have with other assets remaining On-Premises; decisions will be needed regarding the related assets: either move at the same time, or implement solutions enabling the relation to still function after the cutover of the moved application.

Under *Plan* > *Dependency groups*, MPA models these relations through dependency groups. A dependency group is a pool of applications and their resources that can't be migrated without impacting one or more assets in the pool. MPA creates groups by default based on the dependencies the applications have on each other either directly, or through the resources they share. While you have a number of options to control what goes into a group, you don't have a direct say into the content of a group; this is a deliberate design choice as it is intended to ensure handling of the dependencies before finalizing the groups.

First and foremost, you need to have the data: at a minimum, you need a list of business applications, and server to application mappings. You can further add server communication, databases, server and database to application mappings, and application to application dependencies.

Too many relations between the portfolio data can lead to very bulky groups, or just one single group; depending on the number of dependencies created, it can also lead to MPA taking a long time or failing altogether as it tries to compute all the interdependencies. Specifically, server communication can be very chatty and lead to these types of results; look to trim these down either by eliminating certain types of communications, or by removing the servers with most communications – these can often provide common services to all the rest, you can mark them as *Shared infrastructure* on the *Plan* > *Dependency groups* page by selecting *Actions* > *Mark as shared infrastructure*. You can look for these servers also in the server to application mappings.
You can either remove those chatty dependencies by marking them as out-of-scope in *Portfolio data* > *Dependencies*, or remove them altogether on the *Plan* > *Dependency groups* page by selecting *Actions* > *Remove dependency by type*, or disable them one by one by selecting *Actions* > *Remove applications* and marking them for review. Go to *Actions* > *In-Review dependencies* to see all dependencies marked for review and decide if to reinstate them (mark as active) or discard them altogether (mark as inactive).

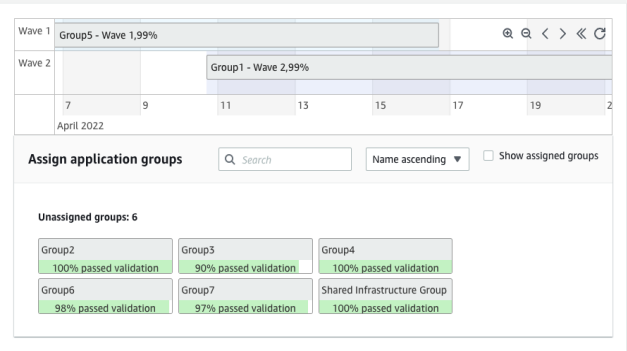
You can view the contents of a group by clicking on the main groups graph on the *Plan* > *Dependency groups* page, of clicking the group name in the table on the same page. On the group's detail page, MPA presents the *Dependency graph* and *Heatmap*. Use the dependency graph for a visual representation of the group, and the heatmap for understanding the applications with the strongest relations in the group.

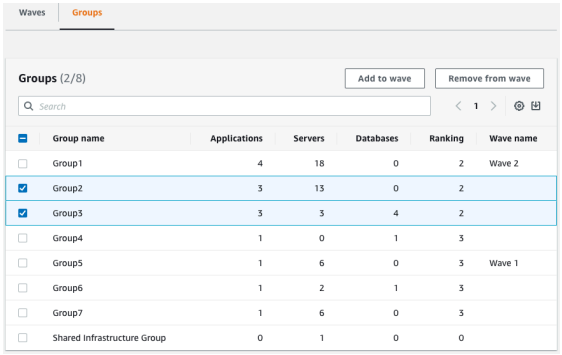
Note:
- Changes in the source data don't automatically reflect into already created groups. You will need to recreate the groups, and all previous groups will be lost.
- The size of a group, based on the number of dependencies between its applications and assets, can lead to the dependencies graph not rendering properly. MPA has set a limit to presenting only up to 1000 relations.

12. Migration wave

Use the *Plan* > *Migration waves* page in MPA to plan the iterations of moving the customer's payload to AWS. A wave consists of one or more dependency groups and is defined by its start and end date. You can define the capacity of a wave in the form of the number of applications, servers and databases that it can include. This allows you to tailor the size of the wave based on some rough idea of the resources available for the actual move.

Create waves either one at a time, or let MPA create the ways by specifying the duration and number of the waves, start and end dates and day of the week to start on.
Once you have waves, allocate dependency groups to them either using drag and drop on the waves timeline, or by selecting the groups from the *Groups* table.



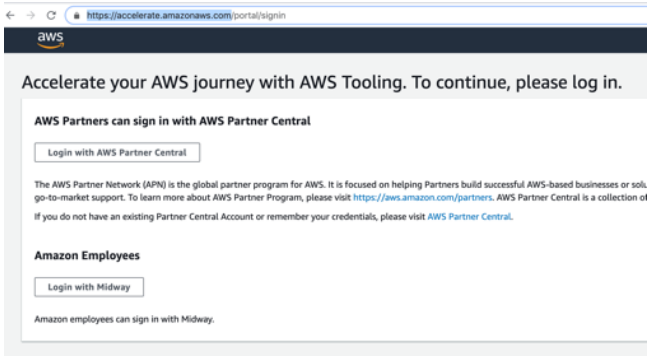


You can download an Excel file with the waves summary by clicking the *Download wave report* button.
The file includes a wave summary, showing the list of waves, their content and timelines. For each of the assets the file also contains separate worksheets with the assets listed by waves and their timeline.

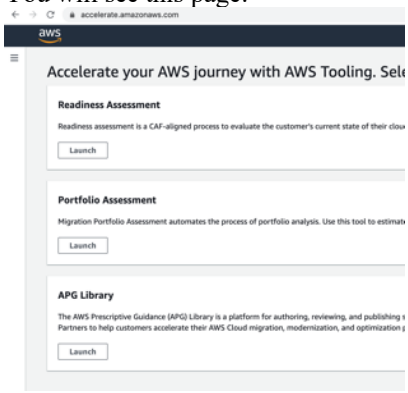
Appendix

1. PDM steps to issue partner access codes

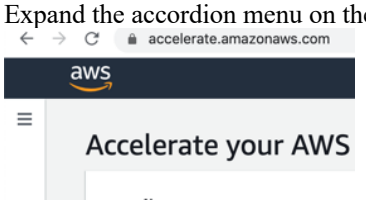
1. Navigate to https://accelerate.amazonaws.com/
   You will see this screen:



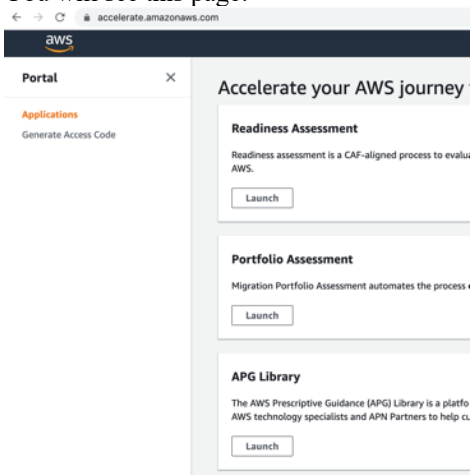
2. Proceed with Login with Midway
   You will see this page:



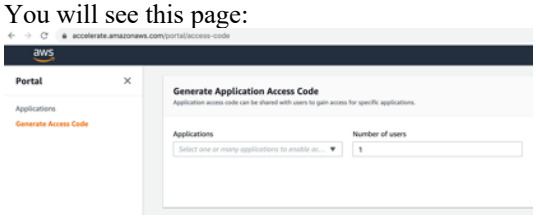
3. Expand the accordion menu on the top left side.



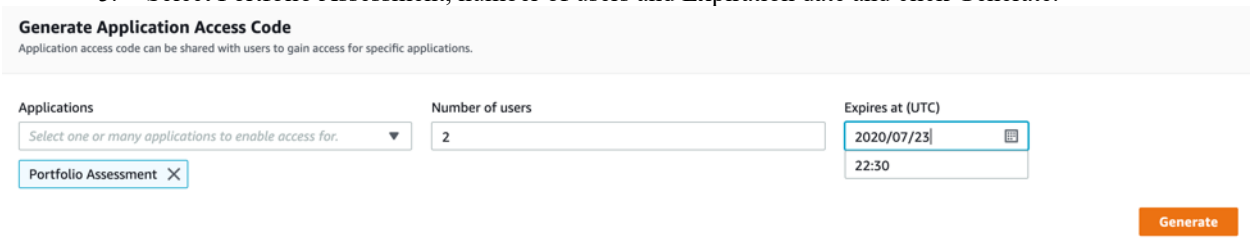
   You will see this page:



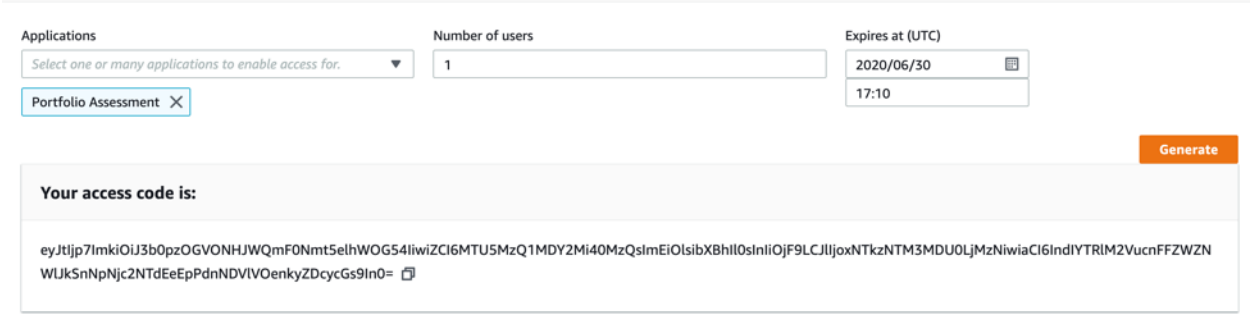
4. Click the Generate Access Code.
   You will see this page:



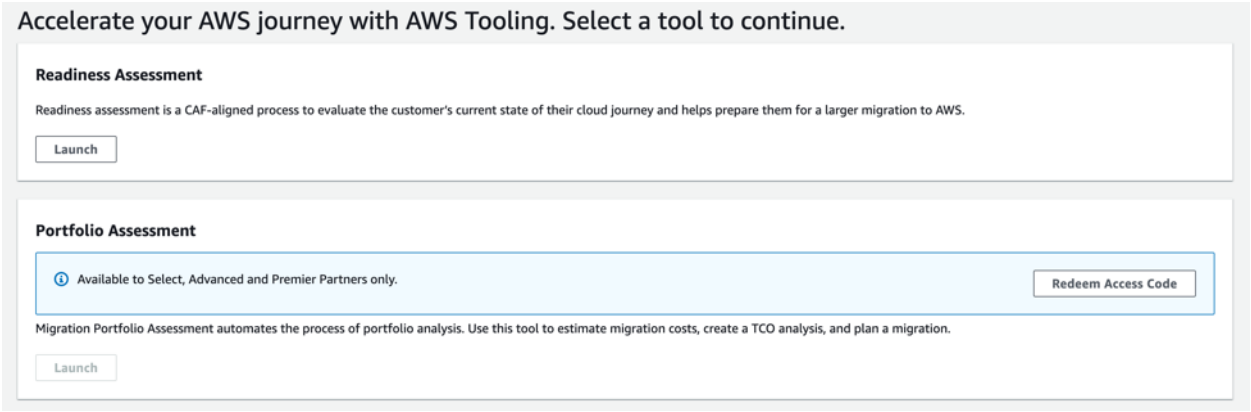
5. Select Portfolio Assessment, number of users and Expiration date and click Generate.



6. The code will be presented on the screen itself for the PDM to copy and share with the partner:



7. The user then needs to access https://accelerate.amazonaws.com/ which presents the option to use the code by clicking *Redeem Access Code*



Document history

Version	Date	Changes
1.0	03/2018	Initial Version
2.0	06/2019	Added details about application grouping, wave planning and prioritization
2.2	05/2020	Added Appendix 1 (PDM steps to issue partner access codes) and reference to the logic MPA uses to determine the R-Types as part of Migration Strategy.
2.3	06/2020	Updated the list of MPA Assumptions.
2.4	07/2020	Updated the partner access code generation process to include partner steps. Updated Gather Customer Portfolio Data to include details about database attributes as well as the threads per core attribute in servers. Updated layout/navigation references to refer the new left nav and tabbed design.
2.5	10/2020	Added RDS and SQL Server Optimization sections.
2.6	03/2021	Added details about the new Data Validation capabilities as well as the existing VMware Cloud on AWS.
2.7	04/2021	Removed the RAID/FTT and updated the number of cores references in the MPA VMC logic, in alignment with the code changes.
2.8	04/2021	Added the EC2 Recommendations section. Added migration cost clarifications and details about support for Glacier.
2.9	10/2021	Added details about importing from ADS. Added dedicated host details. Added application cost details. Added details about Mainframe assessments in MPA.
3.0	05/2022	Reformatted the whole document.
3.1	10/2022	Updated rules for Migration Strategy and EC2 instance preferences.
3.2	11/2022	Updated references to Adjustments following the new implementation.
3.3	04/2023	Added references to EC2 with SQL Server LI
3.4	05/2023	Added references to EC2 shared with Oracle BYOL
3.5	03/2024	Updated following release of Server target architecture features
3.6	06/2024	Removed VMC references following MPA stopping support for VMC recommendations.
3.7	07/2024	Added VMware on-prem logic for multiple products.
3.8	08/2024	Added Network cost for on-prem and AWS. Updated references to EC2 preferences with Graviton.

Thank you!