

Cost Management documentation

Microsoft Cost Management is a suite of FinOps tools that help organizations analyze, monitor and optimize their Microsoft Cloud costs.

Reporting + analytics

OVERVIEW

[Overview](#)

QUICKSTART

[Start analyzing costs](#)

HOW-TO GUIDE

[Analyze unexpected costs](#)

[Use Cost analysis for common tasks](#)

[Connect with Power BI](#)

Monitoring

HOW-TO GUIDE

[Use cost alerts](#)

[Create a budget](#)

[Configure anomaly alerts](#)

[Reservation utilization alerts](#)

[Subscribe to scheduled alerts](#)

Optimization

HOW-TO GUIDE

[Act on recommendations](#)

[Save with savings plans](#)

[Save with reservations](#)

[Manage Azure Hybrid Benefit for SQL](#)

Pricing + estimation

REFERENCE

[Pricing calculator ↗](#)

[TCO calculator ↗](#)

[Azure Migrate](#)

Cost allocation

HOW-TO GUIDE

[Enable tag inheritance](#)

[Split shared costs](#)

Automation + extensibility

OVERVIEW

[Overview](#)

HOW-TO GUIDE

[Choose a cost details solution](#)

[Automation for partners](#)

What is Microsoft Cost Management

Article • 06/16/2023

Microsoft Cost Management is a suite of FinOps tools that help organizations analyze, monitor, and optimize their Microsoft Cloud costs. Cost Management is available to anyone with access to a billing account, subscription, resource group, or management group. You can access Cost Management within the billing and resource management experiences or separately as a standalone tool optimized for FinOps teams who manage cost across multiple scopes. You can also automate and extend native capabilities or enrich your own tools and processes with cost to maximize organizational visibility and accountability with all stakeholders and realize your optimization and efficiency goals faster.

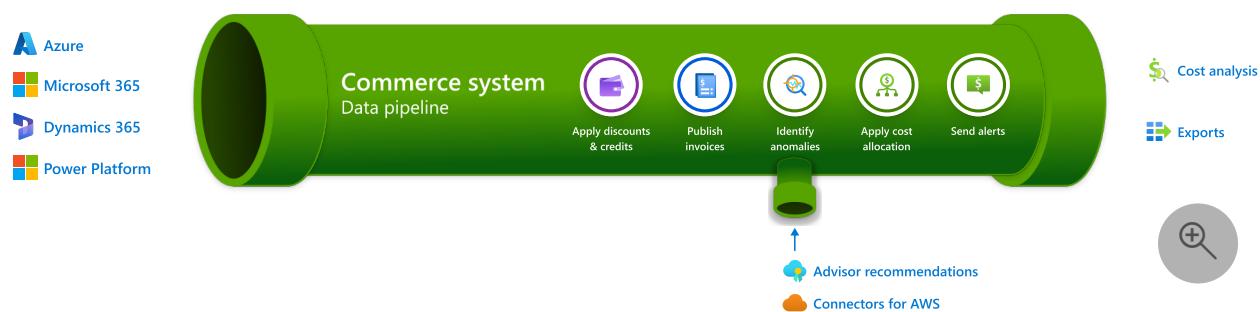
A few examples of what you can do in Cost Management include:

- Report on and analyze costs in the Azure portal, Microsoft 365 admin center, or Power BI.
- Monitor costs proactively with budget, anomaly, reservation utilization, and scheduled alerts.
- Enable tag inheritance and split shared costs with cost allocation rules.
- Automate business processes or integrate cost into external tools by exporting data.

How charges are processed

To understand how Cost Management works, you should first understand the Commerce system. At its core, Microsoft Commerce is a data pipeline that underpins all Microsoft commercial transactions, whether consumer or commercial. While there are many inputs and connections to this pipeline, like the sign-up and Marketplace purchase experiences, this article focuses on the components that help you monitor, allocate, and optimize your costs.

HOW CHARGES ARE PROCESSED IN YOUR BILLING ACCOUNT



From the left, your Azure, Microsoft 365, Dynamics 365, and Power Platform services are all pushing data into the Commerce data pipeline. Each service publishes data on a different cadence. In general, if data for one service is slower than another, it's due to how frequently those services are publishing their usage and charges.

As the data makes its way through the pipeline, the rating system applies discounts based on your specific price sheet and generates "rated usage," which includes price and quantity for each cost record. It's the basis for what you see in Cost Management and it's covered later. At the end of the month, credits are applied and the invoice is published. This process starts 72 hours after your billing period ends, which is usually the last day of the calendar month for most accounts. For example, if your billing period ends on March 31, charges will be finalized on April 4 at midnight.

Important

Credits are applied like a gift card or other payment instrument before the invoice is generated. While credit status is tracked as new charges flow into the data pipeline, credits aren't explicitly applied to these charges until the end of the month.

Everything up to this point makes up the billing process where charges are finalized, discounts are applied, and invoices are published. Billing account and billing profile owners may be familiar with this process as part of the Billing experience within the Azure portal or Microsoft 365 admin center. The Billing experience allows you to review credits, manage your billing address and payment methods, pay invoices, and more – everything related to managing your billing relationship with Microsoft.

- The [anomaly detection](#) model identifies anomalies daily based on normalized usage (not rated usage).
- The cost allocation engine applies tag inheritance and [splits shared costs](#).
- AWS cost and usage reports are pulled based on any [connectors for AWS](#) you may have configured.
- Azure Advisor cost recommendations are pulled in to enable cost savings insights for subscriptions and resource groups.
- Cost alerts are sent out for [budgets](#), [anomalies](#), [scheduled alerts](#), and more based on the configured settings.

Lastly, cost details are made available from [cost analysis](#) in the Azure portal and published to your storage account via [scheduled exports](#).

How Cost Management and Billing relate

[Cost Management](#) is a set of FinOps tools that enable you to analyze, manage, and optimize your costs.

[Billing](#) provides all the tools you need to manage your billing account and pay invoices.

While Cost Management is available from within the Billing experience, Cost Management is also available from every subscription, resource group, and management group in the Azure portal to ensure everyone has full visibility into the costs they're responsible for and can optimize their workloads to maximize efficiency. Cost Management is also available independently to streamline the process for managing cost across multiple billing accounts, subscriptions, resource groups, and/or management groups.



What data is included in Cost Management and Billing?

Within the Billing experience, you can manage all the products, subscriptions, and recurring purchases you use; review your credits and commitments; and view and pay your invoices. Invoices are available online or as PDFs and include all billed charges and any applicable taxes. Credits are applied to the total invoice amount when invoices are generated. This invoicing process happens in parallel to Cost Management data processing, which means Cost Management doesn't include credits, taxes, and some purchases, like support charges in non-MCA accounts.

Classic Cloud Solution Provider (CSP) and sponsorship subscriptions aren't supported in Cost Management. These subscriptions will be supported after they transition to Microsoft Customer Agreement.

For more information about supported offers, what data is included, or how data is refreshed and retained in Cost Management, see [Understand Cost Management data](#).

Estimate your cloud costs

During your cloud journey, there are many tools available to help you understand pricing:

- The [Total Cost of Ownership \(TCO\) calculator](#) should be your first stop if you're curious about how much it would cost to move your existing on-premises infrastructure to the cloud.
- [Azure Migrate](#) is a free tool that helps you analyze your on-premises workloads and plan your cloud migration.
- The [Azure pricing calculator](#) helps you estimate the cost of creating new or expanding existing deployments. In this tool, you're able to explore various configurations of many different Azure services as you identify which SKUs and how much usage keeps you within your desired price range. For more information, see the pricing details for each of the services you use.
- The [Virtual Machine Selector Tool](#) is your one-stop-shop for finding the best VMs for your intended solution.
- The [Azure Hybrid Benefit savings calculator](#) helps you estimate the savings of using your existing Windows Server and SQL Server licenses on Azure.

Report on and analyze costs

Cost Management and Billing include several tools to help you understand, report on, and analyze your invoiced Microsoft Cloud and AWS costs.

- **Cost analysis** is a tool for ad-hoc cost exploration. Get quick answers with lightweight insights and analytics. **Power BI** is an advanced solution to build more extensive dashboards and complex reports or combine costs with other data. Power BI is available for billing accounts and billing profiles.
- **Exports and the Cost Details API** enable you to integrate cost details into external systems or business processes.
- **Connectors for AWS** enable you to ingest your AWS cost details into Azure to facilitate managing Azure and AWS costs together. After configured, the connector also enables other capabilities, like budget and scheduled alerts.

For more information, see [Get started with reporting](#).

Organize and allocate costs

Organizing and allocating costs are critical to ensuring invoices are routed to the correct business units and can be further split for internal billing, also known as *chargeback*. The first step to allocating cloud costs is organizing subscriptions and resources in a way that facilitates natural reporting and chargeback. Microsoft offers the following options to organize resources and subscriptions:

- MCA **billing profiles** and **invoice sections** are used to [group subscriptions into invoices](#). Each billing profile represents a separate invoice that can be billed to a different business unit and each invoice section is segmented separately within those invoices. You can also view costs by billing profile or invoice section in costs analysis.
- EA **departments** and **enrollment accounts** are conceptually similar to invoice sections, as groups of subscriptions, but they aren't represented within the invoice PDF. They're included within the cost details backing each invoice, however. You can also view costs by department or enrollment account in costs analysis.
- **Management groups** also allow grouping subscriptions together, but offer a few key differences:
 - Management group access is inherited down to the subscriptions and resources.
 - Management groups can be layered into multiple levels and subscriptions can be placed at any level.
 - Management groups aren't included in cost details.
 - All historical costs are returned for management groups based on the subscriptions currently within that hierarchy. When a subscription moves, all historical cost moves.
 - Azure Policy supports management groups and they can have rules assigned to automate compliance reporting for your cost governance strategy.
- **Subscriptions and resource groups** are the lowest level at which you can organize your cloud solutions. At Microsoft, every product – sometimes even limited to a single region – is managed within its own subscription. It simplifies cost governance but requires more overhead for subscription management. Most organizations use subscriptions for business units and separating dev/test from production or other environments, then use resource groups for the products. It complicates cost management because resource group owners don't have a way to manage cost across resource groups. On the other hand, it's a straightforward way to understand who's responsible for most resource-based charges. Keep in mind that not all charges come from resources and some don't have resource groups or subscriptions associated with them. It also changes as you move to MCA billing accounts.

- **Resource tags** are the only way to add your own business context to cost details and are perhaps the most flexible way to map resources to applications, business units, environments, owners, etc. For more information, see [How tags are used in cost and usage data](#) for limitations and important considerations.

Once your resources and subscriptions are organized using the subscription hierarchy and have the necessary metadata (tags) to facilitate further allocation, use the following tools in Cost Management to streamline cost reporting:

- **Tag inheritance** simplifies the application of tags by copying subscription and resource group tags down to the resources in cost data. These tags aren't saved on the resources themselves. The change only happens within Cost Management and isn't available to other services, like Azure Policy.
- **Cost allocation** offers the ability to "move" or split shared costs from one subscription, resource group, or tag to another subscription, resource group, or tag. Cost allocation doesn't change the invoice. The goal of cost allocation is to reduce overhead and more accurately report on where charges are ultimately coming from (albeit indirectly), which should drive more complete accountability.

How you organize and allocate costs plays a huge role in how people within your organization can manage and optimize costs. Be sure to plan ahead and revisit your allocation strategy yearly.

Monitor costs with alerts

Cost Management and Billing offer many different types of emails and alerts to keep you informed and help you proactively manage your account and incurred costs.

- **Budget alerts** notify recipients when cost exceeds a predefined cost or forecast amount. Budgets can be visualized in cost analysis and are available on every scope supported by Cost Management. Subscription and resource group budgets can also be configured to notify an action group to take automated actions to reduce or even stop further charges.
- **Anomaly alerts** notify recipients when an unexpected change in daily usage has been detected. It can be a spike or a dip. Anomaly detection is only available for subscriptions and can be viewed within the cost analysis preview. Anomaly alerts can be configured from the cost alerts page.
- **Scheduled alerts** notify recipients about the latest costs on a daily, weekly, or monthly schedule based on a saved cost view. Alert emails include a visual chart representation of the view and can optionally include a CSV file. Views are configured in cost analysis, but recipients don't require access to cost in order to view the email, chart, or linked CSV.

- EA commitment balance alerts are automatically sent to any notification contacts configured on the EA billing account when the balance is 90% or 100% used.
- Invoice alerts can be configured for MCA billing profiles and Microsoft Online Services Program (MOSP) subscriptions. For details, see [View and download your Azure invoice](#).

For more information, see [Monitor usage and spending with cost alerts](#).

Optimize costs

Microsoft offers a wide range of tools for optimizing your costs. Some of these tools are available outside the Cost Management and Billing experience, but are included for completeness.

- There are many [free services](#) available in Azure. Be sure to pay close attention to the constraints. Different services are free indefinitely, for 12 months, or 30 days. Some are free up to a specific amount of usage and some may have dependencies on other services that aren't free.
- [Azure Advisor cost recommendations](#) should be your first stop when interested in optimizing existing resources. Advisor recommendations are updated daily and are based on your usage patterns. Advisor is available for subscriptions and resource groups. Management group users can also see recommendations but they need to select the desired subscriptions. Billing users can only see recommendations for subscriptions they have resource access to.
- [Azure savings plans](#) save you money when you have consistent usage of Azure compute resources. A savings plan can significantly reduce your resource costs by up to 65% from pay-as-you-go prices.
- [Azure reservations](#) help you save up to 72% compared to pay-as-you-go rates by precommitting to specific usage amounts for a set time duration.
- [Azure Hybrid Benefit](#) helps you significantly reduce costs by using on-premises Windows Server and SQL Server licenses or RedHat and SUSE Linux subscriptions on Azure.

For other options, see [Azure benefits and incentives](#).

Next steps

For other options, see [Azure benefits and incentives](#).

Enable preview features in Cost Management Labs

Article • 05/25/2023

Cost Management Labs is an experience in the Azure portal where you can get a sneak peek at what's coming in Cost Management. You can engage directly with us to share feedback and help us better understand how you use the service, so we can deliver more tuned and optimized experiences.

This article explains how to explore preview features and provides a list of the recent previews you might be interested in.

Explore preview features

You can explore preview features from the Cost Management overview.

1. On the Cost Management overview page, select the [Try preview](#) command at the top of the page.
2. From there, enable the features you'd like to use and select **Close** at the bottom of the page.

Cost Management ...

X

New features:



Charts in the cost analysis preview

Show daily or monthly cost over time in the cost analysis preview.



Open config items in the menu

Experimental option to show the selected configuration screen as a nested menu item in the Cost Management menu. Please share feedback.



Change scope from menu

Allow changing scope from the menu for quicker navigation



Streamlined menu

Only show settings in Configuration. Remove Exports and Connectors for AWS from the Cost Management menu.

What would you like next? [Share your ideas.](#)

Having a problem? [Report a bug.](#)

Want to opt out of preview features?

To opt out of preview features, please use [portal.azure.com](#). You can also opt in to more stable preview features there.

[Leave preview portal](#)

Close



3. To see the features enabled, close and reopen Cost Management. You can reopen Cost Management by selecting the link in the notification in the top-right corner.



Reopen Cost Management



Please close and [reopen Cost Management](#) to see changes take effect. Features are only enabled for the current session.

If you're interested in getting preview features even earlier:

1. Navigate to Cost Management.
2. Select **Go to preview portal**.

Or, you can go directly to the [Azure preview portal](#).

It's the same experience as the public portal, except with new improvements and preview features. Every change in Cost Management is available in the preview portal a week before it's in the full Azure portal.

We encourage you to try out the preview features available in Cost Management Labs and share your feedback. It's your chance to influence the future direction of Cost Management. To provide feedback, use the **Report a bug** link in the Try preview menu. It's a direct way to communicate with the Cost Management engineering team.

Remember preview features across sessions

Cost Management now remembers preview features across sessions in the preview portal. Select the preview features you're interested in from the **Try preview** menu and you'll see them enabled by default the next time you visit the portal. There's no need to enable the option – preview features are remembered automatically.

Total KPI tooltip

View more details about what costs are included and not included in the Cost analysis preview. You can enable this option from the Try Preview menu.

The Total KPI tooltip can be enabled from the [Try preview](#) menu in the Azure portal. Use the **How would you rate the cost analysis preview?** option at the bottom of the page to share feedback about the preview.

Cloud Solution Provider (CSP) partners can view a breakdown of costs by customer and subscription in the Cost analysis preview. Note this view is only available for Microsoft Partner Agreement (MPA) billing accounts and billing profiles.

The Customers view can be enabled from the [Try preview](#) menu in the Azure portal. Use the **How would you rate the cost analysis preview?** option at the bottom of the page to share feedback about the preview.

Anomaly detection alerts

Get notified by email when a cost anomaly is detected on your subscription.

Anomaly detection is available for Azure global subscriptions in the cost analysis preview.

Here's an example of a cost anomaly shown in cost analysis:

The screenshot shows the 'Cost Management' dashboard for 'Trey Research Corporate'. The left sidebar includes links for Overview, Access control, Diagnose and solve problems, Cost Management (selected), Cost analysis (preview) (selected), Cost analysis, Cost alerts, Budgets, Advisor recommendations, Billing, and Invoices. The main area displays total costs (\$1,192), average cost per day (\$52.84), and a budget of \$1,200 per month. A callout box highlights a message: 'Daily run rate 143% on Jun 2' with a link to 'See insights'. Below this, a table lists three resources: 'analyticsengine' (Kubernetes service), 'treyanalyticsengine' (SQL server), and 'aks-agentpool-35929937-vms' (Virtual machine scale set). A red box surrounds the 'See insights' link in the message box.

To configure anomaly alerts:

1. Open the cost analysis preview.
2. Navigate to **Cost alerts** and select **Add > Add Anomaly alert**.

The screenshot shows the 'Cost Management' dashboard for 'Trey Research Corporate'. The left sidebar includes links for Overview, Access control, Diagnose and solve problems, Cost Management (selected), and Cost alerts (selected). The main area features search and filter tools ('Search (Ctrl+ /)', 'Manage', 'Refresh', 'Dismiss'), and buttons for 'Add budget' and 'Add anomaly alert'. A red box highlights the 'Add anomaly alert' button. Below it, a section for 'Active alerts' shows a count of 2 with a bell icon. A red box also surrounds the 'Add anomaly alert' button.

For more information about anomaly detection and how to configure alerts, see [Identify anomalies and unexpected changes in cost](#).

Anomaly detection is now available by default in Azure global.

Recent and pinned views in the cost analysis preview

Cost analysis is your tool for interactive analytics and insights. You've seen the addition of new views and capabilities, like anomaly detection, in the cost analysis preview. However, classic cost analysis is still the best tool for quick data exploration with simple filtering and grouping. While these capabilities are coming to the preview, we're introducing a new experience that allows you to select which view you want to start with. Whether that is a preview view, a built-in view, or a custom view you created.

The first time you open the cost analysis preview, you see a list of all views. When you return, you see a list of the recently used views to help you get back to where you left off quicker than ever. You can pin any view or even rename or subscribe to alerts for your saved views.

Recent and pinned views are available by default in the cost analysis preview. Use the [How would you rate the cost analysis preview?](#) option at the bottom of the page to share feedback.

Grouping SQL databases and elastic pools

Get an at-a-glance view of your total SQL costs by grouping SQL databases and elastic pools. They're shown under their parent server in the cost analysis preview. This feature is enabled by default.

Understanding what you're being charged for can be complicated. The best place to start for many people is the [Resources view](#) in the cost analysis preview. It shows resources that are incurring cost. But even a straightforward list of resources can be hard to follow when a single deployment includes multiple, related resources. To help summarize your resource costs, we're trying to group related resources together. So, we're changing cost analysis to show child resources.

Many Azure services use nested or child resources. SQL servers have databases, storage accounts have containers, and virtual networks have subnets. Most of the child resources are only used to configure services, but sometimes the resources have their own usage and charges. SQL databases are perhaps the most common example.

SQL databases are deployed as part of a SQL server instance, but usage is tracked at the database level. Additionally, you might also have charges on the parent server, like for Microsoft Defender for Cloud. To get the total cost for your SQL deployment in classic cost analysis, you need to manually sum up the cost of the server and each individual

database. As an example, you can see the **aepool** elastic pool at the top of the following list and the **treyanalyticsengine** server lower down on the first page. What you don't see is another database even lower in the list. You can imagine how troubling this situation would be when you need the total cost of a large server instance with many databases.

Here's an example showing classic cost analysis where multiple related resource costs aren't grouped.

The screenshot shows the Microsoft Cost Management portal for 'Trey Research Corporate'. The main view displays a summary of costs:

- ACTUAL COST (USD)**: \$927.93
- FORECAST UNAVAILABLE**
- BUDGET: MONTHLY**: \$1,826 /mo

The table below lists individual resources with their details and costs:

Resource	Resource type	Location	Resource group n...	Tags	Cost
> treyanalyticsengine / aepool	Microsoft.Sql/servers	US East	analyticsengine	env:prod org:trey	\$234.92
> aks-agentpool-41981281-vmss	Microsoft.Compute/virtu...	US Central	mc_rg_name_cluster_na...	costcenter:1234 reso...	\$187.24
> asp-mistytower-94ec	App Service plan	US Central	mistytower	costcenter:1234 env:...	\$160.74
> sapmon-vm-133913327cb44f	Microsoft.Compute/virtu...	US East	sapmon-rg-133913327c...	costcenter:1234 env:t...	\$61.87
> defaultworkspace-9ec51cf...	Log Analytics workspace	UK South	defaultresourcegroup-suk	env:prod org:trey	\$50.94
> aks-agentpool-419812aks-agent...	Microsoft.Compute/disks	US Central	mc_rg_name_cluster_na...	costcenter:1234 orch...	\$19.71
> kubernetes	Microsoft.Network/loadB...	US East	mc_analyticsengine_anal...	costcenter:1234 env:...	\$18.60
> kubernetes	Microsoft.Network/loadB...	US Central	mc_rg_name_cluster_na...	costcenter:1234 env:...	\$18.52
> aks-secretagent-37798712-vmss	Microsoft.Compute/virtu...	US East	mc_analyticsengine_anal...	costcenter:1234 aks-...	\$17.78
> aks-agentpool-428500aks-agent...	Microsoft.Compute/disks	US East	mc_analyticsengine_anal...	costcenter:1234 orch...	\$15.92
> aks-agentpool-428500aks-agent...	Microsoft.Compute/disks	US East	mc_analyticsengine_anal...	costcenter:1234 orch...	\$15.92
> aks-agentpool-428500aks-agent...	Microsoft.Compute/disks	US East	mc_analyticsengine_anal...	costcenter:1234 orch...	\$15.92
> treyanalyticsengine	Microsoft.Sql/servers	US East	analyticsengine	env:prod org:trey	\$14.96
> aks-agentpool-419812aks-agent...	Microsoft.Compute/disks	US Central	mc_rg_name_cluster_na...	costcenter:1234 orch...	\$14.18
> analytcsengine	Microsoft.ContainerServi...	US East	analyticsengine	env:prod org:trey ...	\$14.18
> treyvirtualmachine_osdisk_1_307...	Microsoft.Compute/disks	AP East	default-activitylogalerts	env:prod org:trey	\$5.79

In the cost analysis preview, the child resources are grouped together under their parent resource. The grouping shows a quick, at-a-glance view of your deployment and its total cost. Using the same subscription, you can now see all three charges grouped together under the server, offering a one-line summary for your total server costs.

Here's an example showing grouped resource costs with the **Grouping SQL databases and elastic pools** preview option enabled.

The screenshot shows the Azure Cost Management interface for the 'Trey Research Corporate' subscription. The main summary table indicates a total cost of \$927.93 over an average of \$29.93 per day. The 'Cost analysis (preview)' section is selected in the sidebar. The main content area shows 25 resources grouped under the 'treyanalyticsengine' resource group. This grouping includes various Azure services: a SQL server, two SQL elastic pools, three SQL databases, an App Service plan, a Kubernetes service, an Azure Monitor workspace, a Log Analytics workspace, and several disks and load balancers. Each resource is detailed with its name, type, resource group, location, subscription, and tags. A callout bubble in the bottom right corner asks for feedback on the cost analysis preview, with a rating of 4.7 stars.

You might also notice the change in row count. Classic cost analysis shows 53 rows where every resource is broken out on its own. The cost analysis preview only shows 25 rows. The difference is that the individual resources are being grouped together, making it easier to get an at-a-glance cost summary.

In addition to SQL servers, you also see other services with child resources, like App Service, Synapse, and VNet gateways. Each is similarly shown grouped together in the cost analysis preview.

Grouping SQL databases and elastic pools is available by default in the cost analysis preview.

Group related resources in the cost analysis preview

Group related resources, like disks under VMs or web apps under App Service plans, by adding a "cm-resource-parent" tag to the child resources with a value of the parent resource ID. Wait 24 hours for tags to be available in usage and your resources are grouped. Leave feedback to let us know how we can improve this experience further for you.

Some resources have related dependencies that aren't explicit children or nested under the logical parent in Azure Resource Manager. Examples include disks used by a virtual machine or web apps assigned to an App Service plan. Unfortunately, Cost Management

isn't aware of these relationships and can't group them automatically. This experimental feature uses tags to summarize the total cost of your related resources together. You see a single row with the parent resource. When you expand the parent resource, you see each linked resource listed individually with their respective cost.

As an example, let's say you have an Azure Virtual Desktop host pool configured with two VMs. Tagging the VMs and corresponding network/disk resources groups them under the host pool, giving you the total cost of the session host VMs in your host pool deployment. This example gets even more interesting if you want to also include the cost of any cloud solutions made available via your host pool.

Total (USD)	Average	Budget: None (create)				
\$45.66	\$1.52 /day	--				
Showing 7 resources (wvd-test2 is 57% of your total cost) See insights						
Name	Type	Resource group	Location	Subscription	Tags	Total
wvd-test2	Host pool	wvd-test2	US West	Visual Studio Enterprise Subscription	--	\$30.62
wvd-test2	Host pool	wvd-test2	US West	Visual Studio Enterprise Subscription	--	\$11.00
wvd-test2-1_3...	Disk	wvd-test2	US West	Visual Studio Enterprise Subscription	costanalysis-parent: /subscriptions/16	\$0.63
wvd-test2-0_6sdisk_1_7...	Disk	wvd-test2	US West	Visual Studio Enterprise Subscription	costanalysis-parent: /subscriptions/16	\$0.06
wvd-test2-0	Virtual machine	wvd-test2	Intercontinental, US West	Visual Studio Enterprise Subscription	costanalysis-parent: /subscriptions/16	\$0.23
wvd-test2-1	Virtual machine	wvd-test2	US West, Intercontinental	Visual Studio Enterprise Subscription	costanalysis-parent: /subscriptions/16	\$0.13

Before you link resources together, think about how you'd like to see them grouped. You can only link a resource to one parent and cost analysis only supports one level of grouping today.

Once you know which resources you'd like to group, use the following steps to tag your resources:

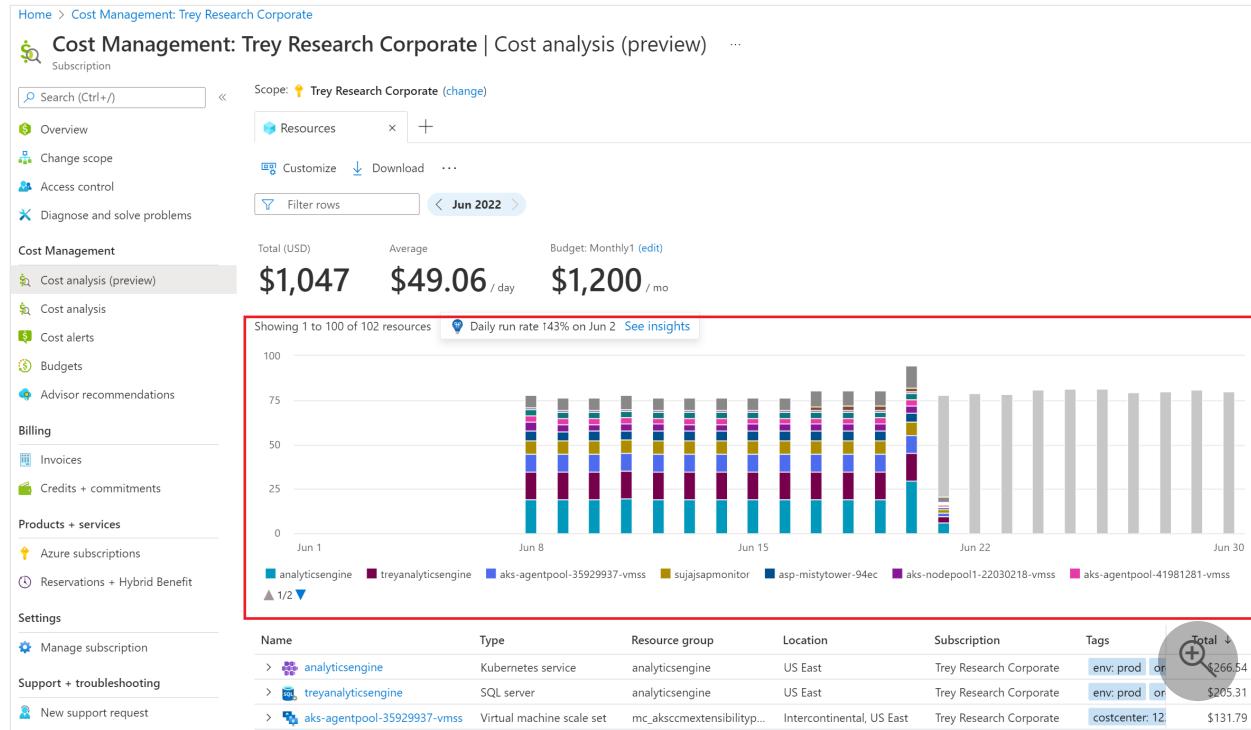
1. Open the resource that you want to be the parent.
2. Select **Properties** in the resource menu.
3. Find the **Resource ID** property and copy its value.
4. Open **All resources** or the resource group that has the resources you want to link.
5. Select the checkboxes for every resource you want to link and then select the **Assign tags** command.
6. Specify a tag key of "cm-resource-parent" (make sure it's typed correctly) and paste the resource ID from step 3.
7. Wait 24 hours for new usage to be sent to Cost Management with the tags. (Keep in mind resources must be actively running with charges for tags to be updated in Cost Management.)
8. Open the [Resources view](#) in the cost analysis preview.

Wait for the tags to load in the Resources view and you should now see your logical parent resource with its linked children. If you don't see them grouped yet, check the tags on the linked resources to ensure they're set. If not, check again in 24 hours.

Grouping related resources is available by default in the cost analysis preview.

Charts in the cost analysis preview

Charts in the cost analysis preview include a chart of daily or monthly charges for the specified date range.



Charts are enabled on the [Try preview](#) page in the Azure portal. Use the **How would you rate the cost analysis preview?** option at the bottom of the page to share feedback about the preview.

Forecast in the cost analysis preview

Show the forecast for the current period at the top of the cost analysis preview.

The Forecast KPI can be enabled from the [Try preview](#) page in the Azure portal. Use the **How would you rate the cost analysis preview?** option at the bottom of the page to share feedback about the preview.

Cost savings insights in the cost analysis preview

Cost insights surface important details about your subscriptions, like potential anomalies or top cost contributors. To support your cost optimization goals, cost insights now include the total cost savings available from Azure Advisor for your subscription.

Cost savings insights are available by default for all subscriptions in the cost analysis preview.

View cost for your resources

Cost analysis is available from every management group, subscription, resource group, and billing scope in the Azure portal and the Microsoft 365 admin center. To make cost data more readily accessible for resource owners, you can now find a **View cost** link at the top-right of every resource overview screen, in **Essentials**. Select the link to open classic cost analysis with a resource filter applied.

The view cost link is enabled by default in the [Azure preview portal](#).

Streamlined menu

Cost Management includes a central management screen for all configuration settings. Some of the settings are also available directly from the Cost Management menu currently. Enabling the **Streamlined menu** option removes configuration settings from the menu.

In the following image, the left menu is classic cost analysis. The right menu is the streamlined menu.

 Overview	 Overview
 Access control	 Change scope
 Diagnose and solve problems	 Access control
Cost Management	
 Cost analysis (preview)	 Cost analysis (preview)
 Cost analysis	 Cost analysis
 Cost alerts	 Cost alerts
 Budgets	 Budgets
 Advisor recommendations	 Advisor recommendations
Billing	
 Invoices	 Invoices
 Credits + commitments	 Credits + commitments
Products + services	
 Azure subscriptions	 Azure subscriptions
 Reservations + Hybrid Benefit	 Reservations + Hybrid Benefit
Settings	
 Manage subscription	 Manage subscription
 Exports	 Exports
 Cost allocation (preview)	 Cost allocation (preview)
 Connectors for AWS	 Connectors for AWS
Support + troubleshooting	
 New support request	

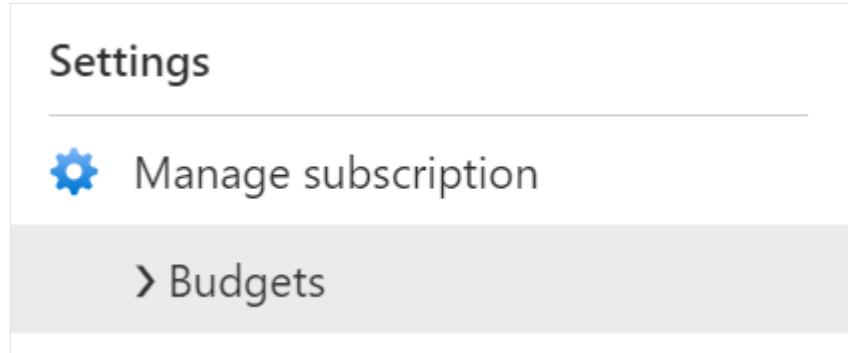
You can enable **Streamlined menu** on the [Try preview](#) page in the Azure portal. Feel free to [share your feedback](#). As an experimental feature, we need your feedback to

determine whether to release or remove the preview.

Open config items in the menu

Cost Management includes a central management view for all configuration settings.

Currently, selecting a setting opens the configuration page outside of the Cost Management menu.



Open config items in the menu is an experimental option to open the configuration page in the Cost Management menu. The option makes it easier to switch to other menu items with one selection. The feature works best with the [streamlined menu](#).

You can enable **Open config items in the menu** on the [Try preview ↗](#) page in the Azure portal.

[Share your feedback ↗](#) about the feature. As an experimental feature, we need your feedback to determine whether to release or remove the preview.

Change scope from menu

If you manage many subscriptions, resource groups, or management groups and need to switch between them often, you might want to include the **Change scope from menu** option.

 **Cost Management: Trey Research Corporate | Cost analysis (preview)** ...

Subscription

Search (Ctrl+ /) <> Scope:  Trey Research Corporate ([change](#))

 Overview  Resources  +

 Access control  Customize  Download ...

 Diagnose and solve problems  Filter rows < Jun 2022 >

Cost Management

Total	Average	Budget: Monthly1 (edit)
 Cost analysis (preview)		\$1,200 / mo
 Cost analysis		
 Cost alerts		
 Budgets		
 Advisor recommendations		

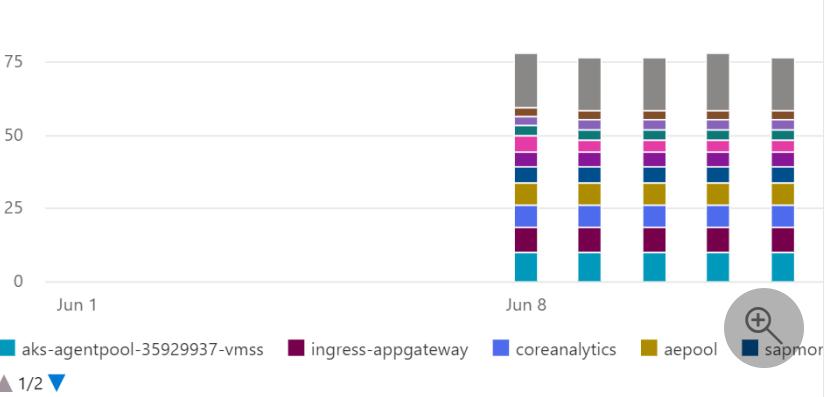
Billing

 Invoices	
 Credits + commitments	

Products + services

 Azure subscriptions	
 Reservations + Hybrid Benefit	

Showing 1 to 100 of 103 resources  Loading...



Legend: aks-agentpool-35929937-vms (blue), ingress-appgatway (purple), coreanalytics (light blue), aepool (yellow), sapmor (dark blue).

It allows changing the scope from the menu for quicker navigation. To enable the feature, navigate to the [Cost Management Labs preview page](#) in the Azure portal.

Share your feedback [about](#) the feature. As an experimental feature, we need your feedback to determine whether to release or remove the preview.

Reservation utilization alerts

Azure reservations can provide cost savings by committing to one-year or three-year plans. However, reservations can sometimes go unutilized or underutilized, resulting in financial losses. As a billing account or reservation user, you can [review the utilization percentage](#) of your reservation purchases in the Azure portal, but you might miss out important changes. By enabling reservation utilization alerts, you solve this by receiving email notifications whenever any of your reservations exhibit low utilization. This allows you to take prompt action and optimize your reservation purchases for maximum efficiency.

The alert email provides essential information including top unutilized reservations and a hyperlink to the list of reservations. By promptly optimizing your reservation purchases, you can avoid financial losses and ensure that your investments are

delivering the expected cost savings. For more information, see [Reservation utilization alerts](#).

The screenshot shows the Azure Cost Management portal. On the left, there's a navigation sidebar with links like Home, Overview, Change scope, Access control, Diagnose and solve problems, Cost Management, Cost analysis, Cost alerts, Budgets, Advisor recommendations, Billing, Usage + charges, Invoices, Credits + commitments, Products + services, Azure subscriptions, Reservations + Hybrid Benefit, Settings, Manage billing account, Support + troubleshooting, and New support request. The main area displays 'Active alerts' with 4 items. Each alert has a checkbox, Type (Budget - cost exceeded threshold), Name, and Date. Below this is a 'Description' section with the text: 'The cost is greater than \$5 and exceeds Test budget threshold of 50%. | Analyze in cost analysis'. To the right, a 'Create alert rule' dialog is open. It includes fields for Condition (Alert type: Reservation utilization, Services: All services selected, Reservations: All reservations selected, Utilization is less than: 95 % for the last 7 days, Start on: 2023-05-09, Sent Weekly until 2024-05-09), Notifications (Recipients: admin@contoso.com, Language: English (United States)), and Information (Alert name: Sample_RUalert_3-3-23). At the bottom are 'Create' and 'Cancel' buttons.

How to share feedback

We're always listening and making constant improvements based on your feedback, so we welcome it. Here are a few ways to share your feedback with the team:

- If you have a problem or are seeing data that doesn't make sense, submit a support request. It's the fastest way to investigate and resolve data issues and major bugs.
- For feature requests, you can share ideas and vote up others in the [Cost Management feedback forum](#).
- Take advantage of the **How would you rate...** prompts in the Azure portal to let us know how each experience is working for you. We monitor the feedback proactively to identify and prioritize changes. You see either a blue option in the bottom-right corner of the page or a banner at the top.

Next steps

Learn about [what's new in Cost Management](#).

Manage AWS costs and usage in Azure

Article • 04/06/2023

After you've set up and configured AWS Cost and Usage report integration for Cost Management, you're ready to start managing your AWS costs and usage. This article helps you understand how to use cost analysis and budgets in Cost Management to manage your AWS costs and usage.

If you haven't already configured the integration, see [Set up and configure AWS Usage report integration](#).

Before you begin: If you're unfamiliar with cost analysis, see the [Explore and analyze costs with Cost analysis](#) quickstart. And, if you're unfamiliar with budgets in Azure, see the [Create and manage budgets](#) tutorial.

View AWS costs in cost analysis

AWS costs are available in Cost Analysis in the following scopes:

- AWS linked accounts under a management group
- AWS linked account costs
- AWS consolidated account costs

The next sections describe how to use the scopes so that you see cost and usage data for each one.

View AWS linked accounts under a management group

Viewing costs by using the management group scope is the only way to see aggregated costs coming from different Azure subscriptions and AWS linked accounts. Using a management group provides a cross-cloud view to view costs from Azure and AWS together.

In cost analysis, open the scope picker and select the management group that holds your AWS linked accounts. Here's an example image in the Azure portal:

Select scope

X

Cost Management + Billing

Scopes are levels in the resource hierarchy where you manage and control access to one or more resources. Select a scope to see a filtered roll-up of all resources, products, and services.

[Learn more](#)

⟨  Root management g...



Select this management group

 Search to filter items...

 test mgt group

 Demo mgt group > ⓘ

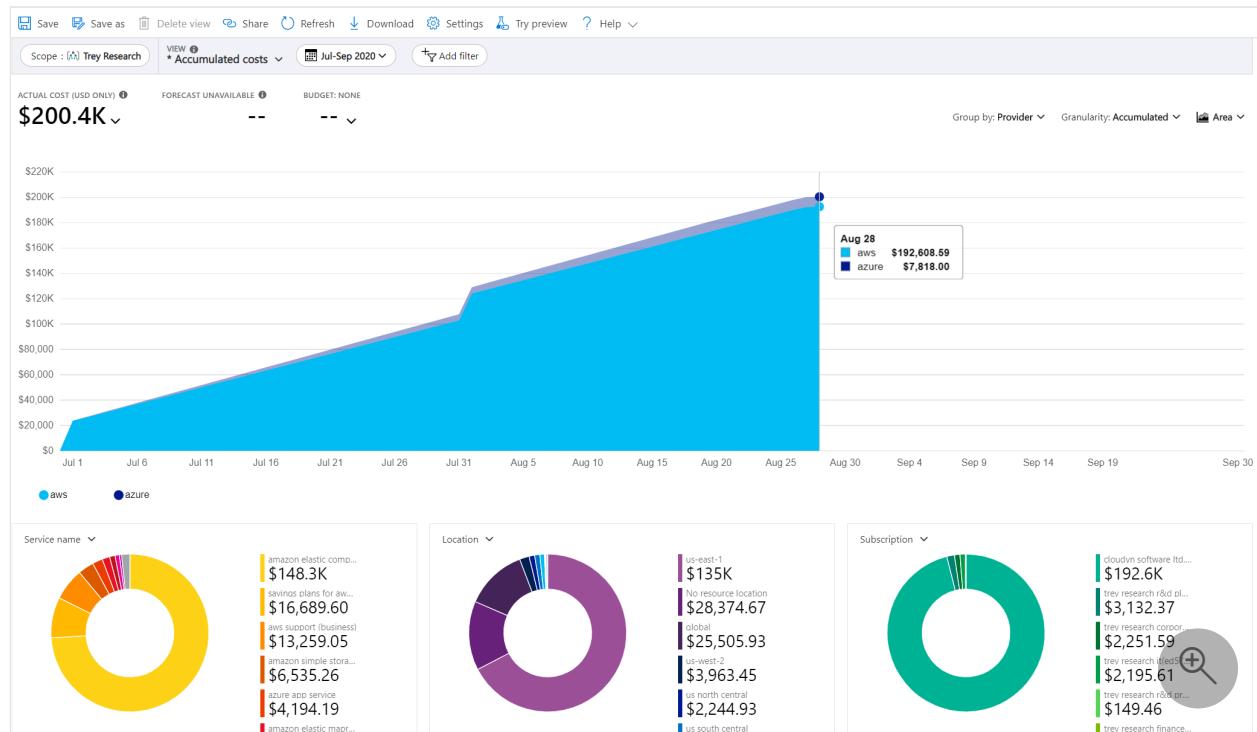
 Cloud + AI Platform ⓘ

 CnAI Orchestration Service Public Corp prod ⓘ

 Trey Research

 Cost Management Research

Here's an example showing the management group cost in cost analysis, grouped by Provider (Azure and AWS).



 Note

Management groups aren't currently supported for Microsoft Customer Agreement (MCA) customers. MCA customers can create the connector and view their AWS data. However, MCA customers can't view their Azure costs and AWS costs together under a management group.

View AWS linked account costs

To view AWS link account costs, open the scope picker and select the AWS linked account. Note that linked accounts are associated to a management group, as defined in the AWS connector.

Here's an example that shows selecting an AWS linked account scope.

The screenshot shows the 'Select scope' dialog box from the Cost Management + Billing service. At the top, it says 'Cost Management + Billing' and has a close button ('X'). Below that, a descriptive text states: 'Scopes are levels in the resource hierarchy where you manage and control access to one or more resources. Select a scope to see a filtered roll-up of all resources, products, and services.' A 'Learn more' link is provided. In the center, there's a breadcrumb navigation: '< Root management g... >' followed by 'Trey Research'. To the right are back and forward navigation icons. A blue button labeled 'Select this management group' is highlighted. Below the button is a search bar with the placeholder 'Search to filter items...'. A list of management groups is shown, with 'Cloudyn Software Ltd.' being the selected item, indicated by a red border around its icon and name. Other items in the list include 'Trey US', 'Trey Worldwide', '657473078308', and 'Babonet'.

View AWS consolidated account costs

To view AWS consolidated account costs, open the scope picker and select the AWS consolidated account. Here's an example that shows selecting an AWS consolidated account scope.

Select scope

X

Cost Management + Billing

Scopes are levels in the resource hierarchy where you manage and control access to one or more resources. Select a scope to see a filtered roll-up of all resources, products, and services.

[Learn more](#)

Microsoft



Current directory – [Switch directories](#)

Search to filter items...



<TestSubscriptionName>



<TestSubscriptionName>



<TestSubscriptionName>



<TestSubscriptionName>



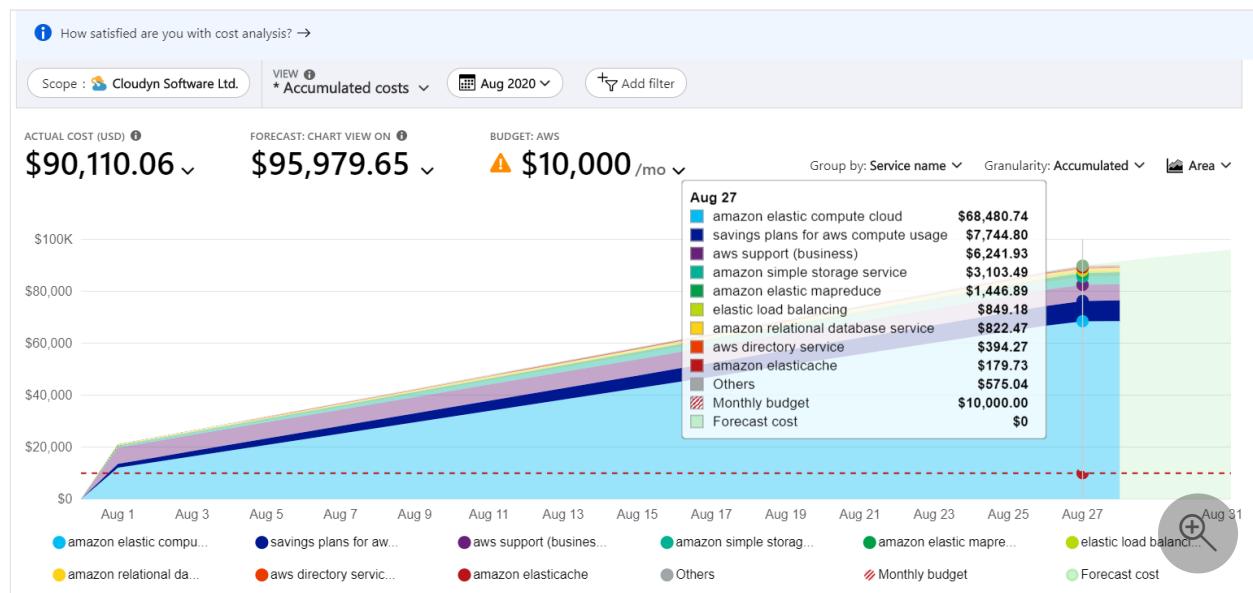
Cloudyn Software Ltd.

Only show subscriptions selected in the global filter. [Change filter](#)

[Select](#)

[Cancel](#)

This scope provides an aggregated view of all AWS linked accounts associated with the AWS consolidated account. Here's an example showing costs for an AWS consolidated account, grouped by service name.



Dimensions available for filtering and grouping

The following table describes dimensions available to group and filter by in cost analysis.

Dimension	Amazon CUR header	Scopes	Comments
Availability zone	lineitem/AvailabilityZone	All	
Location	product/Region	All	
Meter		All	
Meter category	lineItem/ProductCode	All	
Meter subcategory	lineitem/UsageType	All	
Operation	lineItem/Operation	All	
Resource	lineItem/ResourceId	All	
Resource type	product/instanceType	All	If product/instanceType is null, lineItem/UsageType is used.
ResourceGuid	N/A	All	Azure meter GUID.
Service name	product/ProductName	All	If product/ProductName is null, lineItem/ProductCode is used.
Service tier			
Subscription ID	lineItem/UsageAccountId	Consolidated account and management group	
Subscription name	N/A	Consolidated account and management group	Account names are collected using the AWS Organization API.
Tag	resourceTags	All	The <i>user:</i> prefix is removed from user-defined tags to allow cross-cloud tags. The <i>aws:</i> prefix is left intact.
Billing account ID	bill/PayerAccountId	Management group	

Dimension	Amazon CUR header	Scopes	Comments
Billing account name	N/A	Management group	Account names are collected using the AWS Organization API.
Provider	N/A	Management group	Either AWS or Azure.

Set budgets on AWS scopes

Use budgets to proactively manage costs and drive accountability in your organization. Budgets are set on the AWS consolidated account and AWS linked account scopes. Here's an example of budgets for an AWS consolidated account shown in Cost Management:

Name	Scope	Reset period	Creation date	Expiration date	Budget	Evaluated spend	Progress
Monthly	aws-432263259397 (External ...)	Monthly	4/1/2019	3/31/2021	\$125,000.00	\$73,306.68	58.65%
Quarterly	aws-432263259397 (External ...)	Quarterly	4/1/2019	3/31/2021	\$375,000.00	\$159,216.06	42.46%
Yearly	aws-432263259397 (External ...)	Annually	4/1/2019	3/31/2021	\$1,500,000.00	\$386,289.71	25.75%
A100	aws-432263259397 (External ...)	Monthly	5/1/2019	4/30/2021	\$1,000.00	\$73,306.68	100.00%
<BudgetName>	aws-432263259397 (External ...)	Monthly	6/1/2019	5/31/2021	\$12.00	\$75,926.78	100.00%
ForecastedCA	aws-432263259397 (External ...)	Monthly	7/1/2019	6/30/2021	\$300.00	\$85,882.54	100.00%
forecastedMg	aws-432263259397 (External ...)	Monthly	7/1/2019	6/30/2021	\$100.00	\$85,882.54	100.00%
asd	aws-432263259397 (External ...)	Monthly	7/1/2019	6/30/2021	\$30.00	\$75,926.78	100.00%
<BudgetName>	aws-432263259397 (External ...)	Quarterly	7/1/2019	6/30/2021	\$317,688.00	N/A	0.00%
AWS	aws-432263259397 (External ...)	Monthly	1/1/2020	12/31/2021	\$10,000.00	\$75,926.78	100.00%
zzz	aws-432263259397 (External ...)	Monthly	1/1/2020	12/31/2021	\$43,236.00	\$75,926.78	100.00%

AWS data collection process

After setting up the AWS connector, data collection and discovery processes start. It might take few hours to collect all usage data. The duration depends on:

- The time needed to process the CUR files that are in the AWS S3 bucket.
- The time needed to create the AWS Consolidated account and AWS Linked account scopes.
- The time and frequency of AWS are writing the Cost and Usage Report files in the S3 bucket

AWS integration pricing

Each AWS connector gets 90 free trial days.

The list price is 1% of your AWS monthly costs. Each month you are charged based on your invoiced costs from the previous month.

Accessing AWS APIs may incur additional costs on AWS.

AWS integration limitations

- Budgets in Cost Management don't support management groups with multiple currencies. Management groups with multiple currencies won't see a budget evaluation. An error message is shown if you select a management group that has multiple currencies when you create a budget.
- Cloud connectors don't support AWS GovCloud (US), AWS Gov, or AWS China.
- Cost Management shows AWS *usage costs* only. Tax, support, refunds, RI, credits or any other charge types aren't supported yet.

Troubleshooting AWS integration

Use the following troubleshooting information to resolve common problems.

No permission to AWS Linked accounts

Error code: *Unauthorized*

There are two ways to get permissions to access AWS linked accounts costs:

- Get access to the management group that has the AWS Linked accounts.
- Have someone give you permission to the AWS linked account.

By default, the AWS connector creator is the owner of all the objects that the connector created. Including, the AWS consolidated account and the AWS linked account.

In order to be able to Verify the connector settings you will need at least a contributor role, reader can not Verify connector settings

Collection failed with AssumeRole

Error code: *FailedToAssumeRole*

This error means that Cost Management is unable to call the AWS AssumeRole API. This problem can happen because of an issue with the role definition. Verify that the following conditions are true:

- The external ID is the same as the one in the role definition and the connector definition.
- The role type is set to **Another AWS account Belonging to you or 3rd party**.
- The **Require MFA** choice is cleared.
- The trusted AWS account in the AWS Role is 432263259397.

Collection failed with Access Denied - CUR report definitions

Error code: *AccessDeniedReportDefinitions*

This error means that Cost Management is unable to see the Cost and Usage report definitions. This permission is used to validate that the CUR is defined as expected by Cost Management. See [Create a Cost and Usage report in AWS](#).

Collection failed with Access Denied - List reports

Error code: *AccessDeniedListReports*

This error means that Cost Management is unable to list the object in the S3 bucket where the CUR is located. AWS IAM policy requires a permission on the bucket and on the objects in the bucket. See [Create a role and policy in AWS](#).

Collection failed with Access Denied - Download report

Error code: *AccessDeniedDownloadReport*

This error means that Cost Management is unable to access and download the CUR files stored in the Amazon S3 bucket. Make sure that the AWS JSON policy attached to the role resembles the example shown at the bottom of the [Create a role and policy in AWS](#) section.

Collection failed since we did not find the Cost and Usage Report

Error code: *FailedToFindReport*

This error means that Cost Management can't find the Cost and Usage report that was defined in the connector. Make sure it isn't deleted and that the AWS JSON policy attached to the role resembles the example shown at the bottom of the [Create a role and policy in AWS](#) section.

Unable to create or verify connector due to Cost and Usage Report definitions mismatch

Error code: *ReportIsNotValid*

This error relates to the definition of AWS Cost and Usage Report, we require specific settings for this report, see the requirements in [Create a Cost and Usage report in AWS](#).

Internal error when creating connector

Error code: *Create connector - Failed to create connector <ConnectorName>. Reason: Internal error. Please verify that the correct AWS properties were provided.*

This error can occur when your AWS connector and subscription are in different management groups. The AWS connector and subscription need to be in the same management group.

Next steps

- If you haven't already configured your Azure environment with management groups, see [Initial setup of management groups](#).

How to optimize your cloud investment with Cost Management

Article • 12/08/2022

Cost Management gives you the tools to plan for, analyze and reduce your spending to maximize your cloud investment. This document provides you with a methodical approach to cost management and highlights the tools available to you as you address your organization's cost challenges. Azure makes it easy to build and deploy cloud solutions. However, it's important that those solutions are optimized to minimize the cost to your organization. Following the principles outlined in this document and using our tools will help to make sure your organization is prepared for success.

Methodology

Cost management is an organizational problem and should be an ongoing practice that begins before you spend money on cloud resources. To successfully implement cost management and optimize costs, your organization must:

- Be prepared with the proper tools for success
- Be accountable for costs
- Take appropriate action to optimize spending

Three key groups, outlined below, must be aligned in your organization to make sure that you successfully manage costs.

- **Finance** - People responsible for approving budget requests across the organization based on cloud spending forecasts. They pay the corresponding bill and assign costs to various teams to drive accountability.
- **Managers** - Business decision makers in an organization that need to understand cloud spending to find the best spending results.
- **App teams** - Engineers managing cloud resources on a day-to-day basis, developing services to meet the organization's needs. These teams need the flexibility to deliver the most value in their defined budgets.

Key principles

Use the principles outlined below to position your organization for success in cloud cost management.

To learn more, watch the [Cost Management setting up for success](#) video. To watch other videos, visit the [Cost Management YouTube channel](#).
<https://www.youtube-nocookie.com/embed/dVuwITdSAZ4>

Planning

Comprehensive, up-front planning allows you to tailor cloud usage to your specific business requirements. Ask yourself:

- What business problem am I solving?
- What usage patterns do I expect from my resources?

Your answers will help you select the offerings that are right for you. They determine the infrastructure to use and how it's used to maximize your Azure efficiency.

Visibility

When structured well, Cost Management helps you to inform people about the Azure costs they're responsible for or for the money they spend. Azure has services designed to give you insight into *where* your money is spent. Take advantage of these tools. They can help you find resources that are underused, remove waste, and maximize cost-saving opportunities.

Accountability

Attribute costs in your organization to make sure that people responsible are accountable for their team's spending. To fully understand your organization's Azure spending, you should organize your resources to maximize insight into cost attribution. Good organization helps to manage and reduce costs and hold people accountable for efficient spending in your organization.

Optimization

Act to reduce your spending. Make the most of it based on the findings gathered through planning and increasing cost visibility. You might consider purchase and licensing optimizations along with infrastructure deployment changes that are discussed in detail later in this document.

Iteration

Everyone in your organization must engage in the cost management lifecycle. They need to stay involved on an ongoing basis to optimize costs. Be rigorous about this iterative process and make it a key tenet of responsible cloud governance in your organization.



Plan with cost in mind

Before you deploy cloud resources, assess the following items:

- The Azure offer that best meets your needs
- The resources you plan to use
- How much they might cost

Azure provides tools to assist you in the assessment process. The tools can give you a good idea of the investment required to enable your workloads. Then you can select the best configuration for your situation.

Azure onboarding options

The first step in maximizing your experience within Cost Management is to investigate and decide which Azure offer is best for you. Think about how you plan to use Azure in the future. Also consider how you want your billing model configured. Consider the following questions when making your decision:

- How long do I plan to use Azure? Am I testing, or do I plan to build longer-term infrastructure?
- How do I want to pay for Azure? Should I prepay for a reduced price or get invoiced at the end of the month?

To learn more about the various options, visit [How to buy Azure](#). Several of the most common billing models are identified below.

Free ↗

- 12 months of popular free services
- \$200 credit in your billing currency to explore services for 30 days
- 25+ services are always free

Pay as you go ↗

- No minimums or commitments
- Competitive Pricing
- Pay only for what you use
- Cancel anytime

Enterprise Agreement ↗

- Options for up-front Azure Prepayment (previously called monetary commitment)
- Access to reduced Azure pricing

Azure in CSP ↗

- CSP partners are the first point of contact for their customers' needs and the center of the customer relationship
- CSP partners provision new customers, order subscriptions, manage subscriptions, and perform admin tasks on behalf of their customers
- CSP partners bundle services with unique solutions or resell Azure while controlling the pricing, terms and billing

Estimate the cost of your solution

Before you deploy any infrastructure, assess how much your solution will cost. The assessment will help you create a budget for your organization for the workload, up-front. Then you can use a budget over time to benchmark the validity of your initial estimation. And you can compare it with the actual cost of your deployed solution.

Azure pricing calculator

The Azure pricing calculator allows you to mix and match different combinations of Azure services to see an estimate of the costs. You can implement your solution using different ways in Azure - each might influence your overall spending. Thinking early about all of the infrastructure needs of your cloud deployment helps you use the tool most effectively. It can help you get a solid estimate of your estimated spending in Azure.

For more information, see the [Azure pricing calculator](#).

Azure Migrate

Azure Migrate is a service that assesses your organization's current workloads in on-premises datacenters. It gives you insight into what you might need from an Azure replacement solution. First, Migrate analyzes your on-premises machines to determine whether migration is feasible. Then, it recommends VM sizing in Azure to maximize performance. Finally, it also creates a cost estimate for an Azure-based solution.

For more information, see [Azure Migrate](#).

Analyze and manage your costs

Keep informed about how your organization's costs evolve over time. Use the following techniques to properly understand and manage your spending.

Organize resources to maximize cost insights and accountability

A well-planned organizational structure for your Azure billing and resource hierarchies helps to give you a good understanding and control over costs as you create your cloud infrastructure. Watch the video [Setting up entity hierarchies](#) to gain a better understanding of the organizational tools that are available and how to take advantage of them. To watch other videos, visit the [Cost Management YouTube channel](#).

<https://www.youtube-nocookie.com/embed/n3TLRaYJ1NY>

As you evaluate and create a hierarchy that meets your needs, ask yourself the following questions.

Which billing hierarchy is available to me and what are the different scopes that I can use?

Identify the billing arrangement for your organization by determining your Azure offer type. The available scopes for each Azure billing arrangement are documented at [Understand and work with scopes](#).

If I have multiple teams, how should I organize my subscriptions and resource groups?

Creating a subscription or resource group for each team is a common practice. They can help you to differentiate costs and hold teams accountable. However, costs are bound to the subscription or resource group.

If you already have teams with multiple subscriptions, consider grouping the subscriptions into management groups to analyze the costs together. Management groups, subscriptions, and resource groups are all part of the Azure RBAC hierarchy. Use them collectively for access control in your teams.

Resources can span across multiple scopes, especially when they're shared by multiple teams or workloads. Consider identifying resources with tags. Tags are discussed further in the next section.

Do I have Development and Production environments?

Consider creating Dev/Test subscriptions for your development environments to take advantage of reduced pricing. If the workloads span multiple teams or Azure scopes, consider using tags to identify them.

Tag shared resources

Tags are an effective way to understand costs that span across multiple teams and Azure scopes. For example, you might have a resource like an email server that many teams use. You can put a shared resource, like the email server, in a subscription that's dedicated to shared resources or put it in an existing subscription. If you put it in an existing subscription, the subscription owner might not want its cost accruing to their team every month. For this example, you can use a tag to identify the resource as being shared.

Similarly, you might also have web apps or environments, such as Test or Production, that use resources across multiple subscriptions owned by different teams. To better understand the full cost of the workloads, tag the resources that they use. When tags are applied properly, you can apply them as a filter in cost analysis to better understand trends.

After you plan for resource tagging, you can configure an Azure Policy definition to enforce tagging on resources. Watch the [How to review tag policies with Cost Management](#) video to understand the tools available that help you enforce scalable resource tagging. To watch other videos, visit the [Cost Management YouTube channel](#).
<https://www.youtube-nocookie.com/embed/nHQYcYGKuyw>

Use cost analysis

Cost analysis allows you to analyze your organizational costs in-depth by slicing and dicing your costs using standard resource properties. Consider the following common

questions as a guide for your analysis. Answering these questions on a regular basis will help you stay more informed and enable more cost-conscious decisions.

- **Estimated costs for the current month** – How much have I incurred so far this month? Will I stay under my budget?
- **Investigate anomalies** – Do routine checks to make sure that costs stay within a reasonable range of normal usage. What are the trends? Are there any outliers?
- **Invoice reconciliation** – Is my latest invoiced cost more than the previous month? How did spending habits change month-over-month?
- **Internal chargeback** – Now that I know how much I'm being charged, how should those charges be broken down for my organization?

For more information, see [cost analysis](#).

Export billing data on a schedule

Do you need to import your billing data into an external system, like a dashboard or financial system? Set up automated exports to Azure Storage and avoid manually downloading files every month. You can then easily set up automatic integrations with other systems to keep your billing data in sync.

For more information about exporting billing data, see [Create and manage exported data](#).

Create budgets

After you've identified and analyzed your spending patterns, it's important to begin setting limits for yourself and your teams. Budgets give you the ability to set either a cost or usage-based budget with many thresholds and alerts. Make sure to review the budgets that you create regularly to see your budget burn-down progress and make changes as needed. Budgets also allow you to configure an automation trigger when a given budget threshold is reached. For example, you can configure your service to shut down VMs. Or you can move your infrastructure to a different pricing tier in response to a budget trigger.

For more information, see [Create budgets](#).

For more information about budget-based automation, see [Budget Based Automation](#).

Act to optimize

Use the following ways to optimize spending.

Cut out waste

After you've deployed your infrastructure in Azure, it's important to make sure it is being used. The easiest way to start saving immediately is to review your resources and remove any that aren't being used. From there, you should determine if your resources are being used as efficiently as possible.

Azure Advisor

Azure Advisor is a service that, among other things, identifies virtual machines with low utilization from a CPU or network usage standpoint. From there, you can decide to either shut down or resize the machine based on the estimated cost to continue running the machines. Advisor also provides recommendations for reserved instance purchases. The recommendations are based on your last 30 days of virtual machine usage. When acted on, the recommendations can help you reduce your spending.

For more information, see [Azure Advisor](#).

Size your VMs properly

VM sizing has a significant impact on your overall Azure cost. The number of VMs needed in Azure might not equate to what you currently have deployed in an on-premises datacenter. Make sure you choose the right size for the workloads that you plan to run.

For more information, see [Azure IaaS: proper sizing and cost](#).

Use purchase discounts

Azure has many discounts that your organization should take advantage of to save money.

Azure savings plan for compute

Azure savings plan for compute is our most flexible savings plan. It lets you save up to 65 percent on pay-as-you-go prices and applies to a broad range of compute services across subscriptions, resource groups, management groups or entire Azure accounts. You select an hourly compute commitment for a one-year or three-year term. The longer the commitment, the more savings you earn. You can pay monthly for no additional cost, and Azure automatically applies the largest savings to your account.

For more information, see [Azure savings plan for compute](#).

Azure Reservations

Azure Reservations allow you to prepay for one-year or three-years of virtual machine or SQL Database compute capacity. Pre-paying will allow you to get a discount on the resources you use. Azure reservations can significantly reduce your virtual machine or SQL database compute costs — up to 72 percent on pay-as-you-go prices with one-year or three-year upfront commitment. Reservations provide a billing discount and don't affect the runtime state of your virtual machines or SQL databases.

For more information, see [What are Azure Reservations?](#).

Use Azure Hybrid Benefit

If you already have Windows Server or SQL Server licenses in your on-premises deployments, you can use the Azure Hybrid Benefit program to save in Azure. With the Windows Server benefit, each license covers the cost of the OS (up to two virtual machines), and you only pay for base compute costs. You can use existing SQL Server licenses to save up to 55 percent on vCore-based SQL Database options. Options include SQL Server in Azure Virtual Machines and SQL Server Integration Services.

For more information, see [Azure Hybrid Benefit savings calculator](#).

Other resources

Azure also has a service that allows you to build services that take advantage of surplus capacity in Azure for reduced rates. For more information, see [Use low priority VMs with Batch](#).

Next steps

- If you're new to Cost Management, read [What is Cost Management?](#) to learn how it helps monitor and control Azure spending and to optimize resource use.

Understand Cost Management data

Article • 02/24/2023

This article helps you better understand Azure cost and usage data that's included in Cost Management. It explains how frequently data is processed, collected, shown, and closed. You're billed for Azure usage monthly. Although billing cycles are monthly periods, cycle start and end dates vary by subscription type. How often Cost Management receives usage data varies based on different factors. Such factors include how long it takes to process the data and how frequently Azure services emit usage to the billing system.

Cost Management includes all usage and purchases, including reservations and third-party offerings for Enterprise Agreement (EA) accounts. Microsoft Customer Agreement accounts and individual subscriptions with pay-as-you-go rates only include usage from Azure and Marketplace services. Support and other costs aren't included. Costs are estimated until an invoice is generated and don't factor in credits. Cost Management also includes costs associated with New Commerce products like Microsoft 365 and Dynamics 365 that are invoiced along with Azure. Currently, only Partners can purchase New Commerce non-Azure products.

If you have a new subscription, you can't immediately use Cost Management features. It might take up to 48 hours before you can use all Cost Management features.

Supported Microsoft Azure offers

The following information shows the currently supported [Microsoft Azure offers](#) in Cost Management. An Azure offer is the type of the Azure subscription that you have. Data is available in Cost Management starting on the **Data available from** date. Summarized data in cost analysis is only available for the last 13 months. If a subscription changes offers, costs before the offer change date aren't available.

Category	Offer name	Quota ID	Offer number	Data available from
Azure Government	Azure Government Enterprise	EnterpriseAgreement_2014-09-01	MS-AZR-USGOV-0017P	May 2014 ¹

Category	Offer name	Quota ID	Offer number	Data available from
Azure Government	Azure Government Pay-As-You-Go	PayAsYouGo_2014-09-01	MS-AZR-USGOV-0003P	October 2, 2018
Enterprise Agreement (EA)	Enterprise Dev/Test	MSDNDevTest_2014-09-01	MS-AZR-0148P	May 2014 ¹
Enterprise Agreement (EA)	Microsoft Azure Enterprise	EnterpriseAgreement_2014-09-01	MS-AZR-0017P	May 2014 ¹
Microsoft Customer Agreement	Microsoft Azure Plan	EnterpriseAgreement_2014-09-01	N/A	March 2019 ²
Microsoft Customer Agreement	Microsoft Azure Plan for Dev/Test	MSDNDevTest_2014-09-01	N/A	March 2019 ²
Microsoft Customer Agreement supported by partners	Microsoft Azure Plan	CSP_2015-05-01, CSP_MG_2017-12-01, and CSPDEVTEST_2018-05-01 ⁴	N/A	October 2019
Microsoft Developer Network (MSDN)	MSDN Platforms ³	MSDN_2014-09-01	MS-AZR-0062P	October 2, 2018
Pay-As-You-Go	Pay-As-You-Go	PayAsYouGo_2014-09-01	MS-AZR-0003P	October 2, 2018
Pay-As-You-Go	Pay-As-You-Go Dev/Test	MSDNDevTest_2014-09-01	MS-AZR-0023P	October 2, 2018
Pay-As-You-Go	Microsoft Cloud Partner Program	MPN_2014-09-01	MS-AZR-0025P	October 2, 2018
Pay-As-You-Go	Free Trial ³	FreeTrial_2014-09-01	MS-AZR-0044P	October 2, 2018
Pay-As-You-Go	Azure in Open ³	AzureInOpen_2014-09-01	MS-AZR-0111P	October 2, 2018

Category	Offer name	Quota ID	Offer number	Data available from
Pay-As-You-Go	Azure Pass ³	AzurePass_2014-09-01	MS-AZR-0120P, MS-AZR-0122P - MS-AZR-0125P, MS-AZR-0128P - MS-AZR-0130P	October 2, 2018
Visual Studio	Visual Studio Enterprise – MPN ³	MPN_2014-09-01	MS-AZR-0029P	October 2, 2018
Visual Studio	Visual Studio Professional ³	MSDN_2014-09-01	MS-AZR-0059P	October 2, 2018
Visual Studio	Visual Studio Test Professional ³	MSDNDevTest_2014-09-01	MS-AZR-0060P	October 2, 2018
Visual Studio	Visual Studio Enterprise ³	MSDN_2014-09-01	MS-AZR-0063P	October 2, 2018
Visual Studio	Visual Studio Enterprise: BizSpark ³	MSDN_2014-09-01	MS-AZR-0064P	October 2, 2018

¹ For data before May 2014, visit the [Azure Enterprise portal](#).

² Microsoft Customer Agreements started in March 2019 and don't have any historical data before this point.

³ Historical data for credit-based and pay-in-advance subscriptions might not match your invoice. See [Historical data may not match invoice](#) below.

⁴ Quota IDs are the same across Microsoft Customer Agreement and classic subscription offers. Classic CSP subscriptions are not supported.

The following offers aren't supported yet:

Category	Offer name	Quota ID	Offer number
Azure Germany	Azure Germany Pay-As-You-Go	PayAsYouGo_2014-09-01	MS-AZR-DE-0003P
Cloud Solution Provider (CSP)	Microsoft Azure	CSP_2015-05-01	MS-AZR-0145P

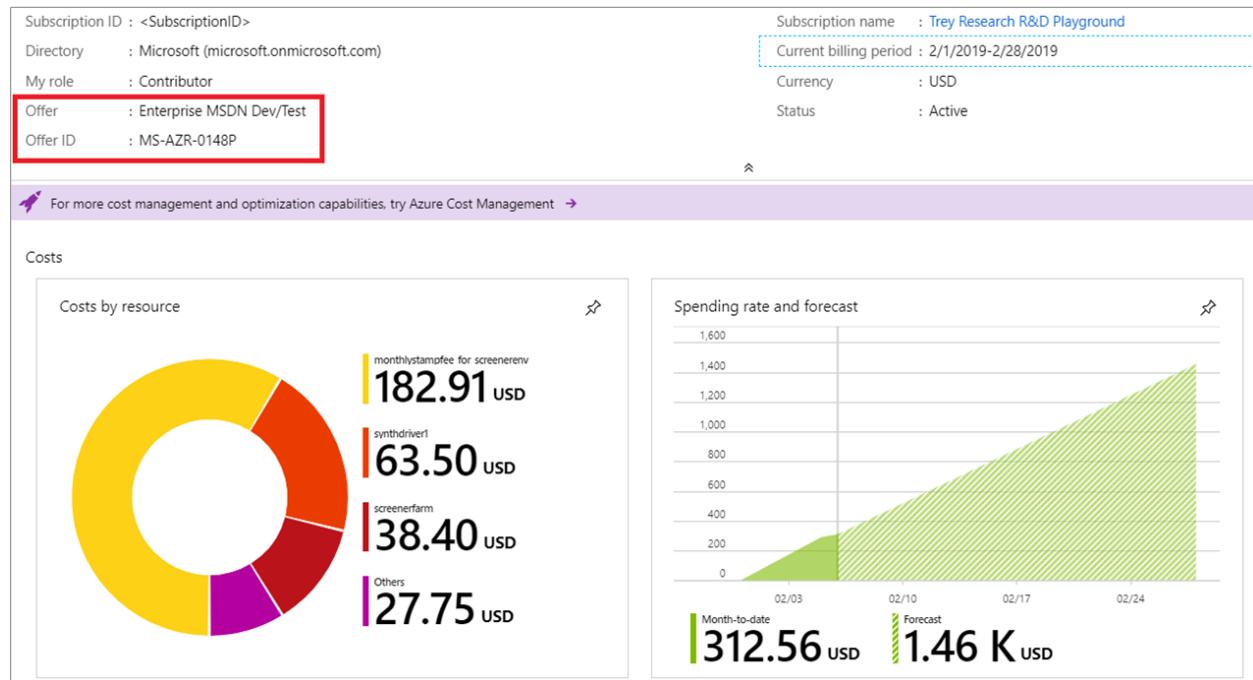
Category	Offer name	Quota ID	Offer number
Cloud Solution Provider (CSP)	Azure Government CSP	CSP_2015-05-01	MS-AZR-USGOV-0145P
Cloud Solution Provider (CSP)	Azure Germany in CSP for Microsoft Cloud Germany	CSP_2015-05-01	MS-AZR-DE-0145P
Pay-As-You-Go	Azure for Students Starter	DreamSpark_2015-02-01	MS-AZR-0144P
Pay-As-You-Go	Azure for Students ³	AzureForStudents_2018-01-01	MS-AZR-0170P
Pay-As-You-Go	Microsoft Azure Sponsorship	Sponsored_2016-01-01	MS-AZR-0036P
Support Plans	Standard support	Default_2014-09-01	MS-AZR-0041P
Support Plans	Professional Direct support	Default_2014-09-01	MS-AZR-0042P
Support Plans	Developer support	Default_2014-09-01	MS-AZR-0043P
Support Plans	Germany support plan	Default_2014-09-01	MS-AZR-DE-0043P
Support Plans	Azure Government Standard Support	Default_2014-09-01	MS-AZR-USGOV-0041P
Support Plans	Azure Government Pro-Direct Support	Default_2014-09-01	MS-AZR-USGOV-0042P
Support Plans	Azure Government Developer Support	Default_2014-09-01	MS-AZR-USGOV-0043P

Free trial to pay-as-you-go upgrade

For information about the availability of free tier services after you upgrade to pay-as-you-go pricing from a Free trial, see the [Azure free account FAQ](#).

Determine your offer type

If you don't see data for a subscription and you want to determine if your subscription falls under the supported offers, you can validate that your subscription is supported. To validate an Azure subscription is supported, sign in to the Azure portal. Then select **All Services** in the left menu pane. In the list of services, select **Subscriptions**. In the subscription list menu, select the subscription that you want to verify. Your subscription is shown on the Overview tab and you can see the **Offer** and **Offer ID**. The following image shows an example.



Costs included in Cost Management

The following tables show data that's included or isn't in Cost Management. All costs are estimated until an invoice is generated. Costs shown don't include free and prepaid credits.

Included	Not included
Azure service usage ⁵	Support charges - For more information, see Invoice terms explained .
Marketplace offering usage ⁶	Taxes - For more information, see Invoice terms explained .
Marketplace purchases ⁶	Credits - For more information, see Invoice terms explained .
Reservation purchases ⁷	
Amortization of reservation purchases ⁷	

Included	Not included
New Commerce non-Azure products (Microsoft 365 and Dynamics 365) ⁸	

⁵ Azure service usage is based on reservation and negotiated prices.

⁶ Marketplace purchases aren't available for MSDN and Visual Studio offers at this time.

⁷ Reservation purchases are only available for Enterprise Agreement (EA) and Microsoft Customer Agreement accounts at this time.

⁸ Only available for specific offers.

How tags are used in cost and usage data

Cost Management receives tags as part of each usage record submitted by the individual services. The following constraints apply to these tags:

- Tags must be applied directly to resources and aren't implicitly inherited from the parent resource group.
- Resource tags are only supported for resources deployed to resource groups.
- Some deployed resources may not support tags or may not include tags in usage data.
- Resource tags are only included in usage data while the tag is applied – tags aren't applied to historical data.
- Resource tags are only available in Cost Management after the data is refreshed.
- Resource tags are only available in Cost Management when the resource is active/running and producing usage records. For example, when a VM is deallocated.
- Managing tags requires contributor access to each resource or the [tag contributor](#) RBAC role.
- Managing tag policies requires either owner or policy contributor access to a management group, subscription, or resource group.

If you don't see a specific tag in Cost Management, consider the following questions:

- Was the tag applied directly to the resource?
- Was the tag applied more than 24 hours ago?
- Does the resource type support tags? Some resource types don't support tags in usage data. See [Tags support for Azure resources](#) for the full list of what is supported.

Here are a few tips for working with tags:

- Plan ahead and define a tagging strategy that allows you to break down costs by organization, application, environment, and so on.
- [Group and allocate costs using tag inheritance](#) to apply resource group and subscription tags to child resource usage records. If you were using Azure policy to enforce tagging for cost reporting, consider enabling the tag inheritance setting for easier management and more flexibility.
- Use the Tags API with either Query or UsageDetails to get all cost based on the current tags.

Cost and usage data updates and retention

Cost and usage data is typically available in Cost Management within 8-24 hours. Keep the following points in mind as you review costs:

- Each Azure service (such as Storage, Compute, and SQL) emits usage at different intervals – You might see data for some services sooner than others.
- Estimated charges for the current billing period are updated six times per day.
- Estimated charges for the current billing period can change as you incur more usage.
- Each update is cumulative and includes all the line items and information from the previous update.
- Azure finalizes or *closes* the current billing period up to 72 hours (three calendar days) after the billing period ends.
- During the open month (uninvoiced) period, cost management data should be considered an estimate only. In some cases, charges may be latent in arriving to the system after the usage actually occurred.

The following examples illustrate how billing periods could end:

- Enterprise Agreement (EA) subscriptions – If the billing month ends on March 31, estimated charges are updated up to 72 hours later. In this example, by midnight (UTC) April 4.
- Pay-as-you-go subscriptions – If the billing month ends on May 15, then the estimated charges might get updated up to 72 hours later. In this example, by midnight (UTC) May 19.

After your billing period ends and your invoice is created, it can take up to 48 hours later for the usage data to get finalized. If the usage file isn't ready, you'll see a message on the Invoices page in the Azure portal stating `Your usage and charges file is not ready`. After the usage file is available, you can download it.

Once cost and usage data becomes available in Cost Management, it will be retained for at least seven years. Only the last 13 months are available from the portal. For historical data before 13 months, please use [Exports](#) or the [Cost Details API](#).

Rerated data

Whether you use the Cost Management APIs, Power BI, or the Azure portal to retrieve data, expect the current billing period's charges to get rerated. Charges might change until the invoice is closed.

Cost rounding

Costs shown in Cost Management are rounded. Costs returned by the Query API aren't rounded. For example:

- Cost analysis in the portal - Charges are rounded using standard rounding rules: values more than 0.5 and higher are rounded up, otherwise costs are rounded down. Rounding occurs only when values are shown. Rounding doesn't happen during data processing and aggregation. For example, cost analysis aggregates costs as follows:
 - Charge 1: \$0.004
 - Charge 2: \$0.004
 - Aggregate charge rendered: $0.004 + 0.004 = 0.008$. The charge shown is \$0.01.
- Query API - Charges are shown at eight decimal places and rounding doesn't occur.

Historical data might not match invoice

Historical data for credit-based and pay-in-advance offers might not match your invoice. Some Azure pay-as-you-go, MSDN, and Visual Studio offers can have Azure credits and advanced payments applied to the invoice. The historical data shown in Cost Management is based on your estimated consumption charges only. Cost Management historical data doesn't include payments and credits. Historical data shown for the following offers may not match exactly with your invoice.

- Azure for Students (MS-AZR-0170P)
- Azure in Open (MS-AZR-0111P)
- Azure Pass (MS-AZR-0120P, MS-AZR-0123P, MS-AZR-0125P, MS-AZR-0128P, MS-AZR-0129P)
- Free Trial (MS-AZR-0044P)

- MSDN (MS-AZR-0062P)
- Visual Studio (MS-AZR-0029P, MS-AZR-0059P, MS-AZR-0060P, MS-AZR-0063P, MS-AZR-0064P)

Next steps

- If you haven't already completed the first quickstart for Cost Management, read it at [Start analyzing costs](#).

Understand and work with scopes

Article • 05/10/2023

This article helps you understand billing and resource management scopes available in Azure and how to use the scopes in Cost Management and APIs.

Scopes

A *scope* is a node in the Azure resource hierarchy where Azure AD users access and manage services. Most Azure resources are created and deployed into resource groups, which are part of subscriptions. Microsoft also offers two hierarchies above Azure subscriptions that have specialized roles to manage billing data:

- Billing data, such as payments and invoices
- Cloud services, such as cost and policy governance

Scopes are where you manage billing data, have roles specific to payments, view invoices, and conduct general account management. Billing and account roles are managed separately from roles used for resource management, which use [Azure RBAC](#). To clearly distinguish the intent of the separate scopes, including the access control differences, they're referred to as *billing scopes* and *Azure RBAC scopes*, respectively.

To learn more about scopes, watch the [Cost Management setting up hierarchies](#) video. To watch other videos, visit the [Cost Management YouTube channel](#).

<https://www.youtube-nocookie.com/embed/n3TLRaYJ1NY>

How Cost Management uses scopes

Cost Management works at all scopes above resources to allow organizations to manage costs at the level at which they have access, whether that's the entire billing account or a single resource group. Although billing scopes differ based on your Microsoft agreement (subscription type), the Azure RBAC scopes don't.

Azure RBAC scopes

Azure supports three scopes for resource management. Each scope supports managing access and governance, including but not limited to, cost management.

- [Management groups](#) - Hierarchical containers, used to organize Azure subscriptions. A management group tree can support up to six levels of depth. The

limit doesn't include the Root level or the subscription level.

Resource type: [Microsoft.Management/managementGroups](#)

- **Subscriptions** - Primary containers for Azure resources.

Resource type: [Microsoft.Resources/subscriptions](#)

- **Resource groups** - Logical groupings of related resources for an Azure solution that share the same lifecycle. For example resources that are deployed and deleted together.

Resource type: [Microsoft.Resources/subscriptions/resourceGroups](#)

Management groups allow you to organize subscriptions into a hierarchy. For example, you might create a logical organization hierarchy using management groups. Then, give teams subscriptions for production and dev/test workloads. And then create resource groups in the subscriptions to manage each subsystem or component.

Creating an organizational hierarchy allows cost and policy compliance to roll up organizationally. Then, each leader can view and analyze their current costs. And then they can create budgets to curb bad spending patterns and optimize costs with Advisor recommendations at the lowest level.

Granting access to view costs and optionally manage cost configuration, such as budgets and exports, is done on governance scopes using Azure RBAC. You use Azure RBAC to grant Azure AD users and groups access to do a predefined set of actions. The actions are defined in a role on a specific scope and lower. For instance, a role assigned to a management group scope also grants the same permissions to nested subscriptions and resource groups.

Cost Management supports the following built-in roles for each of the following scopes:

- **Owner** – Can view costs and manage everything, including cost configuration.
- **Contributor** – Can view costs and manage everything, including cost configuration, but excluding access control.
- **Reader** – Can view everything, including cost data and configuration, but can't make any changes.
- **Cost Management Contributor** – Can view costs, manage cost configuration, and view recommendations.
- **Cost Management Reader** – Can view cost data, cost configuration, and view recommendations.

Cost Management Contributor is the recommended least-privilege role. The role allows people to create and manage budgets and exports to more effectively monitor and

report on costs. Cost Management Contributors might also require more roles to support complex cost management scenarios. Consider the following scenarios:

- **Reporting on resource usage** – Cost Management shows cost in the Azure portal. It includes usage as it pertains to cost in the full usage patterns. This report can also show API and download charges, but you may also want to drill into detailed usage metrics in Azure Monitor to get a deeper understanding. Consider granting [Monitoring Reader](#) on any scope where you also need to report detailed usage metrics.
- **Act when budgets are exceeded** – Cost Management Contributors also need access to create and manage action groups to automatically react to overages. Consider granting [Monitoring Contributor](#) to a resource group that contains the action group to use when budget thresholds are exceeded. Automating specific actions requires more roles for the specific services used, such as Automation and Azure Functions.
- **Schedule cost data export** – Cost Management Contributors also need access to manage storage accounts to schedule an export to copy data into a storage account. Consider granting [Storage Account Contributor](#) to a resource group that contains the storage account where cost data is exported.
- **Viewing cost-saving recommendations** – Cost Management Readers and Cost Management Contributors have access to view cost recommendations by default. However, access to act on the cost recommendations requires access to individual resources. Consider granting a [service-specific role](#) if you want to act on a cost-based recommendation.

 **Note**

Management groups aren't currently supported in Cost Management features for Microsoft Customer Agreement subscriptions. The [Cost Details API](#) also doesn't support management groups for either EA or MCA customers.

Management groups are only supported if they contain up to 3,000 Enterprise Agreement (EA), Pay-as-you-go (PAYG), or Microsoft internal subscriptions. Management groups with more than 3,000 subscriptions or subscriptions with other offer types, like Microsoft Customer Agreement or Azure Active Directory subscriptions, can't view costs.

If you have a mix of subscriptions, move the unsupported subscriptions to a separate arm of the management group hierarchy to enable Cost Management for the supported subscriptions. As an example, create two management groups under the root management group: **Azure AD** and **My Org**. Move your Azure AD subscription to the

Azure AD management group and then view and manage costs using the My Org management group.

Feature behavior for each role

The following table shows how Cost Management features are used by each role. The following behavior is applicable to all Azure RBAC scopes.

Feature/Role	Owner	Contributor	Reader	Cost Management Reader	Cost Management Contributor
Cost Analysis / Forecast / Query / Cost Details API	Read only				
Shared views	Create, Read, Update, Delete	Create, Read, Update, Delete	Read only	Read only	Create, Read, Update, Delete
Budgets	Create, Read, Update, Delete	Create, Read, Update, Delete	Read only	Read only	Create, Read, Update, Delete
Alerts	Read, Update	Read, Update	Read only	Read only	Read, Update
Exports	Create, Read, Update, Delete	Create, Read, Update, Delete	Read only	Read only	Create, Read, Update, Delete
Cost Allocation rules	Feature not available for Azure RBAC scopes				

Enterprise Agreement scopes

Enterprise Agreement (EA) billing accounts, also called enrollments, have the following scopes:

- **Billing account** - Represents an EA enrollment. Invoices are generated at this scope. Purchases that aren't usage-based, such as Marketplace and reservations, are only available at this scope. They aren't represented in departments or enrollment accounts. Reservation usage, along with all other usage, is applied to individual resources. Usage rolls-up to subscriptions within the billing account. To

see reservation costs broken down to each resource, switch to view **Amortized cost** in cost analysis.

Resource type: `Microsoft.Billing/billingAccounts (accountType = Enrollment)`

- **Department** - Optional grouping of enrollment accounts.

Resource type: `Billing/billingAccounts/departments`

- **Enrollment account** - Represents a single account owner. Doesn't support granting access to multiple people.

Resource type: `Microsoft.Billing/billingAccounts/enrollmentAccounts`

Although governance scopes are bound to a single directory, EA billing scopes aren't. An EA billing account may have subscriptions across any number of Azure AD directories.

EA billing scopes support the following roles:

- **Enterprise admin** – Can manage billing account settings and access, can view all costs, and can manage cost configuration. For example, budgets and exports.
- **Enterprise read-only user** – Can view billing account settings, cost data, and cost configuration. Can manage budgets and exports.
- **Department admin** – Can manage department settings, such as cost center, and can access, view all costs, and manage cost configuration. For example, budgets and exports. The **DA view charges** billing account setting must be enabled for department admins and read-only users to see costs. If **DA view charges** option is disabled, department users can't see costs at any level, even if they're an account or subscription owner.
- **Department read-only user** – Can view department settings, cost data, and cost configuration. Can manage budgets and exports. If **DA view charges** option is disabled, department users can't see costs at any level, even if they're an account or subscription owner.
- **Account owner** – Can manage enrollment account settings (such as cost center), view all costs, and manage cost configuration (such as budgets and exports) for the enrollment account. The **AO view charges** billing account setting must be enabled for account owners and Azure RBAC users to see costs.

EA billing account users don't have direct access to invoices. Invoices are available from an external volume licensing system.

Azure subscriptions are nested under enrollment accounts. Billing users have access to cost data for the subscriptions and resource groups that are under their respective

scopes. They don't have access to see or manage resources in the Azure portal. Users can view costs by navigating to **Cost Management + Billing** in the Azure portal list of services. Then, they can filter costs to the specific subscriptions and resource groups they need to report on.

Billing users don't have access to management groups because they don't fall explicitly under a specific billing account. Access must be granted to management groups explicitly. Management groups roll-up costs from all nested subscriptions. However, they only include usage-based purchases. They don't include purchases such as reservations and third-party Marketplace offerings. To view these costs, use the EA billing account.

Feature behavior for each role

The following tables show how Cost Management features can be utilized by each role.

Enrollment scope

Feature/Role	Enterprise Admin	Enterprise Read-Only
Cost Analysis / Forecast / Query / Cost Details API	Read only	Read only
Shared Views	Create, Read, Update, Delete	Create, Read, Update, Delete
Budgets/Reservation utilization alerts	Create, Read, Update, Delete	Create, Read, Update, Delete
Alerts	Read, Update	Read, Update
Exports	Create, Read, Update, Delete	Create, Read, Update, Delete
Cost Allocation Rules	Create, Read, Update, Delete	Create, Read, Update, Delete

Department scope

Feature/Role	Enterprise Admin	Enterprise Read Only	Department Admin (only if "DA view charges" setting is on)	Department Read Only (only if "DA view charges" setting is on)

Feature/Role	Enterprise Admin	Enterprise Read Only	Department Admin (only if "DA view charges" setting is on)	Department Read Only (only if "DA view charges" setting is on)
Cost Analysis / Forecast / Query / Cost Details API	Read only	Read only	Read only	Read only
Shared Views	Create, Read, Update, Delete	Create, Read, Update, Delete	Create, Read, Update, Delete	Create, Read, Update, Delete
Budgets	Create, Read, Update, Delete	Create, Read, Update, Delete	Create, Read, Update, Delete	Create, Read, Update, Delete
Alerts	Read, Update	Read, Update	Read, Update	Read, Update
Exports	Create, Read, Update, Delete	Create, Read, Update, Delete	Create, Read, Update, Delete	Create, Read, Update, Delete
Cost Allocation Rules	N/A – only applicable to Billing Account scope	N/A – only applicable to Billing Account scope	N/A – only applicable to Billing Account scope	N/A – only applicable to Billing Account scope

Account scope

Feature/Role	Enterprise Admin	Enterprise Read Only	Department Admin (only if "DA view charges" is on)	Department Read Only (only if "DA view charges" setting is on)	Account Owner (only if "AO view charges" setting is on)
Cost Analysis / Forecast / Query / Cost Details API	Read only	Read only	Read only	Read only	Read only
Shared Views	Create, Read, Update, Delete	Create, Read, Update, Delete	Create, Read, Update, Delete	Create, Read, Update, Delete	Create, Read, Update, Delete

Feature/Role	Enterprise Admin	Enterprise Read Only	Department Admin (only if "DA view charges" is on)	Department Read Only (only if "DA view charges" setting is on)	Account Owner (only if "AO view charges" setting is on)
Budgets	Create, Read, Update, Delete	Create, Read, Update, Delete	Create, Read, Update, Delete	Create, Read, Update, Delete	Create, Read, Update, Delete
Alerts	Read, Update	Read, Update	Read, Update	Read, Update	Read, Update
Exports	Create, Read, Update, Delete	Create, Read, Update, Delete	Create, Read, Update, Delete	Create, Read, Update, Delete	Create, Read, Update, Delete
Cost Allocation Rules	N/A – only applicable to Billing Account scope	N/A – only applicable to Billing Account scope	N/A – only applicable to Billing Account scope	N/A – only applicable to Billing Account scope	N/A – only applicable to Billing Account scope

Individual agreement scopes

Azure subscriptions created from individual offers like pay-as-you-go and related types like Free Trial and dev/test offers, don't have an explicit billing account scope. Instead, each subscription has an account owner or account admin, like the EA account owner.

- **Billing account** - Represents a single account owner for one or more Azure subscriptions. It doesn't currently support granting access to multiple people or access to aggregated cost views.

Resource type: Not applicable

Individual Azure subscription account admins can view and manage billing data, such as invoices and payments, from the [Azure portal](#) > **Subscriptions** > select a subscription.

Unlike EA, individual Azure subscription account admins can see their invoices in the Azure portal. Keep in mind that Cost Management Reader and Cost Management Contributor roles don't provide access to invoices. For more information, see [How to grant access to invoices](#).

Microsoft Customer Agreement scopes

Microsoft Customer Agreement billing accounts have the following scopes:

- **Billing account** - Represents a customer agreement for multiple Microsoft products and services. Customer Agreement billing accounts aren't functionally the same as EA enrollments. EA enrollments are more closely aligned to billing profiles.

Resource type: `Microsoft.Billing/billingAccounts (accountType = Organization)`

- **Billing profile** - Defines the subscriptions that are included in an invoice. Billing profiles are the functional equivalent of an EA enrollment, since that's the scope that invoices are generated at. Similarly, purchases that aren't usage-based (such as Marketplace and reservations) are only available at this scope. They aren't included in invoice sections.

Resource type: `Microsoft.Billing/billingAccounts/billingProfiles`

- **Invoice section** - Represents a group of subscriptions in an invoice or billing profile. Invoice sections are like departments—multiple people can have access to an invoice section.

Resource type: `Microsoft.Billing/billingAccounts/invoiceSections`

- **Customer** - Represents a group of subscriptions that are associated to a specific customer that is onboarded to a Microsoft Customer Agreement by partner. This scope is specific to Cloud Solution Providers (CSP).

Unlike EA billing scopes, Customer Agreement billing accounts *are* managed by a single directory. Microsoft Customer Agreement billing accounts can have *linked* subscriptions that could be in different Azure AD directories.

Customer Agreement billing scopes don't apply to partners. Partner roles and permissions are documented at [Assign users roles and permissions](#).

Customer Agreement billing scopes support the following roles:

- **Owner** – Can manage billing settings and access, view all costs, and manage cost configuration. For example, budgets and exports. In function, this Customer Agreement billing scope is the same as the [Cost Management Contributor Azure role](#).
- **Contributor** – Can manage billing settings except access, view all costs, and manage cost configuration. For example, budgets and exports. In function, this

Customer Agreement billing scope is the same as the [Cost Management Contributor Azure role](#).

- **Reader** – Can view billing settings, cost data, and cost configuration. Can manage budgets and exports.
- **Invoice manager** – Can view and pay invoices and can view cost data and configuration. Can manage budgets and exports.
- **Azure subscription creator** – Can create Azure subscriptions, view costs, and manage cost configuration. For example, budgets and exports. In function, this Customer Agreement billing scope is the same as the EA enrollment account owner role.

Azure subscriptions are nested under invoice sections, like how they are under EA enrollment accounts. Billing users have access to cost data for the subscriptions and resource groups that are under their respective scopes. However, they don't have access to see or manage resources in the Azure portal. Billing users can view costs by navigating to **Cost Management + Billing** in the Azure portal list of services. Then, filter costs to the specific subscriptions and resource groups they need to report on.

 **Note**

Management group scopes aren't supported for Microsoft Customer Agreement accounts at this time.

Billing users don't have access to management groups because they don't explicitly fall under the billing account. However, when management groups are enabled for the organization, all subscription costs are rolled-up to the billing account and to the root management group because they're both constrained to a single directory.

Management groups only include purchases that are usage-based. Purchases like reservations and third-party Marketplace offerings aren't included in management groups. So, the billing account and root management group may report different totals. To view these costs, use the billing account or respective billing profile.

Feature behavior for each role

The following tables show how Cost Management features can be utilized by each role.

Billing account

Feature/Role	Owner	Contributor	Reader
--------------	-------	-------------	--------

Feature/Role	Owner	Contributor	Reader
Cost Analysis / Forecast / Query / Cost Details API	Read only	Read only	Read only
Shared Views	Create, Read, Update, Delete	Create, Read, Update, Delete	Create, Read, Update, Delete
Budgets	Create, Read, Update, Delete	Create, Read, Update, Delete	Create, Read, Update, Delete
Alerts	Read, Update	Read, Update	Read, Update
Exports	Create, Read, Update, Delete	Create, Read, Update, Delete	Create, Read, Update, Delete
Cost Allocation Rules	Create, Read, Update, Delete	Create, Read, Update, Delete	Read only

Billing profile

Feature/Role	Owner	Contributor	Reader	Invoice Manager
Cost Analysis / Forecast / Query / Cost Details API	Read only	Read only	Read only	Read only
Shared Views	Create, Read, Update, Delete			
Budgets/Reservation utilization alerts	Create, Read, Update, Delete			
Alerts	Read, Update	Read, Update	Read, Update	Create, Read, Update, Delete
Exports	Create, Read, Update, Delete	Create, Read, Update, Delete	Create, Read, Update, Delete	Read, Update
Cost Allocation Rules	N/A – only applicable to Billing Account			

Invoice section

Feature/Role	Owner	Contributor	Reader	Azure Subscription Creator
Cost Analysis / Forecast / Query / Cost Details API	Read only	Read only	Read only	Read only
Shared Views	Create, Read, Update, Delete			
Budgets	Create, Read, Update, Delete			
Alerts	Read, Update	Read, Update	Read, Update	Read, Update
Exports	Create, Read, Update, Delete			
Cost Allocation Rules	N/A – only applicable to Billing Account			

AWS scopes

After AWS integration is complete, see [setup and configure AWS integration](#). The following scopes are available:

- **External Billing account** - Represents a customer agreement with a third-party vendor. It's similar to the EA billing account.

Resource type: `Microsoft.CostManagement/externalBillingAccounts`

- **External subscription** - Represents a customer operational account with a third-party vendor. It's similar to an Azure subscription.

Resource type: `Microsoft.CostManagement/externalSubscriptions`

Cloud Solution Provider (CSP) scopes

The following scopes are supported for CSPs with customers on a Microsoft Customer Agreement:

- **Billing account** - Represents a customer agreement for multiple Microsoft products and services. Customer Agreement billing accounts aren't functionally the same as EA enrollments. EA enrollments are more closely aligned to billing profiles.

Resource type: Microsoft.Billing/billingAccounts (accountType = Organization)

- **Billing profile** - Defines the subscriptions that are included in an invoice. Billing profiles are the functional equivalent of an EA enrollment, since that's the scope that invoices are generated at. Similarly, purchases that aren't usage-based (such as Marketplace and reservations) are only available at this scope.

Resource type: Microsoft.Billing/billingAccounts/billingProfiles

- **Customer** - Represents a group of subscriptions that are associated to a specific customer that is onboarded to a Microsoft Customer Agreement by a partner.

Only the users with *Global admin* and *Admin agent* roles can manage and view costs for billing accounts, billing profiles, and customers directly in the partner's Azure tenant. For more information about partner center roles, see [Assign users roles and permissions](#).

Cost Management only supports CSP partner customers if the customers have a Microsoft Customer Agreement. For CSP supported customers who aren't yet on a Microsoft Customer Agreement, see [Partner Center](#).

Management groups in CSP scopes aren't supported by Cost Management. If you have a CSP subscription and you set the scope to a management group in cost analysis, an error similar the following one is shown:

Management group <ManagementGroupName> does not have any valid subscriptions

Switch between scopes in Cost Management

All Cost Management views in the Azure portal include a **Scope** selection pill at the top-left of the view. Use it to quickly change scope. Select the **Scope** pill to open the scope picker. It shows billing accounts, the root management group, and any subscriptions that aren't nested under the root management group. To select a scope, select the background to highlight it, and then select **Select** at the bottom. To drill-in to nested scopes, like resource groups in a subscription, select the scope name link. To select the parent scope at any nested level, select **Select this <scope>** at the top of the scope picker.

View historical billing scopes after migration or contract change

If you migrated from an EA agreement to a Microsoft Customer Agreement, you still have access to your old billing scope.

1. Sign in to the [Azure portal](#).
2. Search for and then select **Cost Management + Billing**.
3. Select **Billing Scope** to view your new and previous billing accounts.

Identify the resource ID for a scope

When you work with Cost Management APIs, knowing the scope is critical. Use the following information to build the proper scope URI for Cost Management APIs.

Billing accounts

1. Open the Azure portal and then navigate to **Cost Management + Billing** in the list of services.
2. Select **Properties** in the billing account menu.
3. Copy the billing account ID.
4. Your scope is:

```
/providers/Microsoft.Billing/billingAccounts/{billingAccountId}"
```

Billing profiles

1. Open the Azure portal and then navigate to **Cost Management + Billing** in the list of services.
2. Select **Billing profiles** in the billing account menu.
3. Select the name of the billing profile.
4. Select **Properties** in the billing profile menu.
5. Copy the billing account and billing profile IDs.
6. Your scope is:

```
/providers/Microsoft.Billing/billingAccounts/{billingAccountId}/billingProfiles/{billingProfileId}"
```

Invoice sections

1. Open the Azure portal and then navigate to **Cost Management + Billing** in the list of services.
2. Select **Invoice sections** in the billing account menu.
3. Select the name of the invoice section.
4. Select **Properties** in the invoice section menu.
5. Copy the billing account and invoice section IDs.
6. Your scope is:

```
/providers/Microsoft.Billing/billingAccounts/{billingAccountId}/invoiceSections/{invoiceSectionId}"
```

```
ns/{invoiceSectionId}"
```

EA departments

1. Open the Azure portal and then navigate to **Cost Management + Billing** in the list of services.
2. Select **Departments** in the billing account menu.
3. Select the name of the department.
4. Select **Properties** in the department menu.
5. Copy the billing account and department IDs.
6. Your scope is:

```
"/providers/Microsoft.Billing/billingAccounts/{billingAccountId}/departments/{  
departmentId}"
```

EA enrollment account

1. Open the Azure portal and navigate to **Cost Management + Billing** in the list of services.
2. Select **Enrollment accounts** in the billing account menu.
3. Select the name of the enrollment account.
4. Select **Properties** in the enrollment account menu.
5. Copy the billing account and enrollment account IDs.
6. Your scope is:

```
"/providers/Microsoft.Billing/billingAccounts/{billingAccountId}/enrollmentAcc  
ounts/{enrollmentAccountId}"
```

Management group

1. Open the Azure portal and navigate to **Management groups** in the list of services.
2. Navigate to the management group.
3. Copy the management group ID from the table.
4. Your scope is: `"/providers/Microsoft.Management/managementGroups/{id}"`

Subscription

1. Open the Azure portal and navigate to **Subscriptions** in the list of services.
2. Copy the subscription ID from the table.
3. Your scope is: `"/subscriptions/{id}"`

Resource groups

1. Open the Azure portal and navigate to **Resource groups** in the list of services.
2. Select the name of the resource group.
3. Select **Properties** in the resource group menu.
4. Copy the resource ID field value.
5. Your scope is: `"/subscriptions/{id}/resourceGroups/{name}"`

Cost Management is currently supported in Azure Global with

<https://management.azure.com> and Azure Government with

<https://management.usgovcloudapi.net>. For more information about Azure Government,

see [Azure Global and Government API endpoints](#).

Next steps

- If you haven't already completed the first quickstart for Cost Management, read it at [Start analyzing costs](#).

Get started with Cost Management for partners

Article • 04/05/2023

Cost Management is natively available for direct partners who have onboarded their customers to a Microsoft Customer Agreement and have [purchased an Azure Plan](#). This article explains how partners use [Cost Management](#) features to view costs for subscriptions in the Azure Plan. It also describes how partners enable Cost Management access at retail rates for their customers.

For direct partners and indirect providers, the global admin and admin agents, can access Cost Management in the partner tenant and manage costs at invoiced prices.

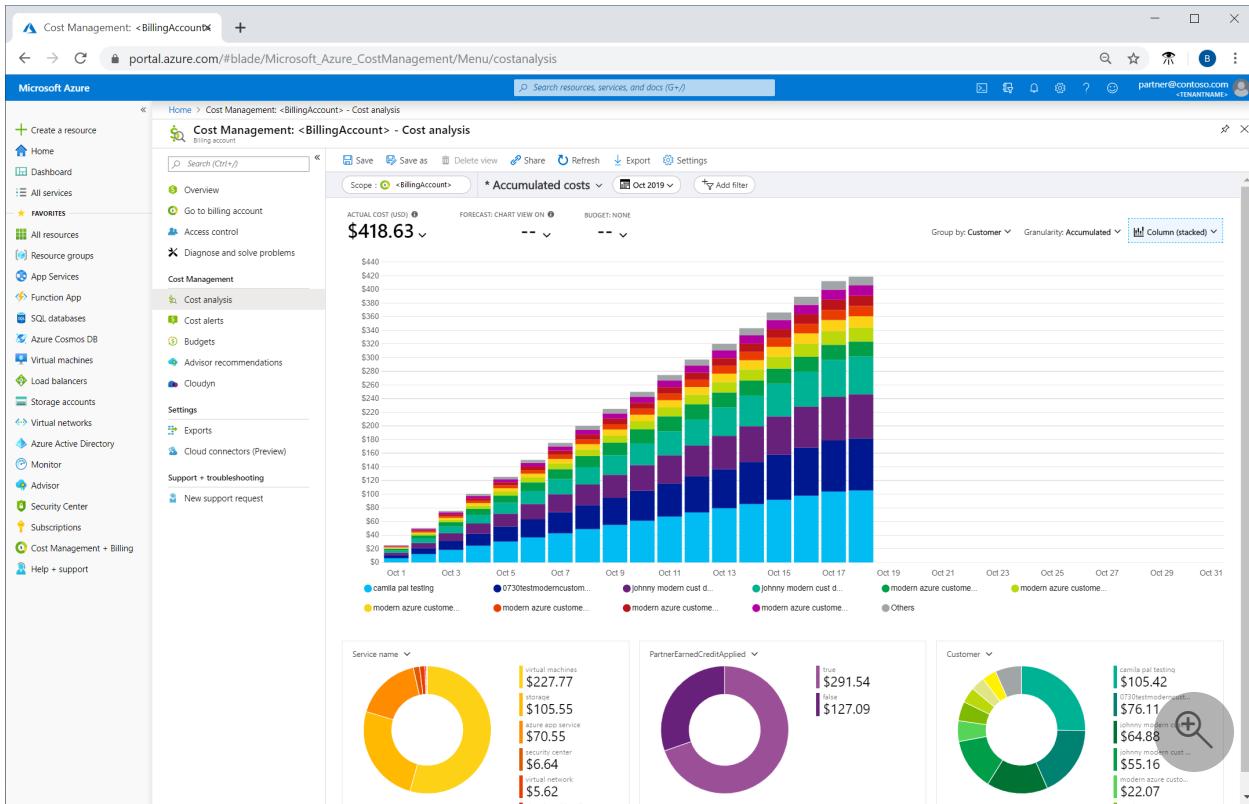
Resellers and customers can access Cost Management in the customer tenant and view consumption costs for each individual subscription, where costs are computed and shown at retail rates. However, they must have Azure RBAC access to the subscription in the customer tenant to view costs. The cost visibility policy must be enabled by the provider for the customer tenant.

Customers can use Cost Management features when enabled by their CSP partner.

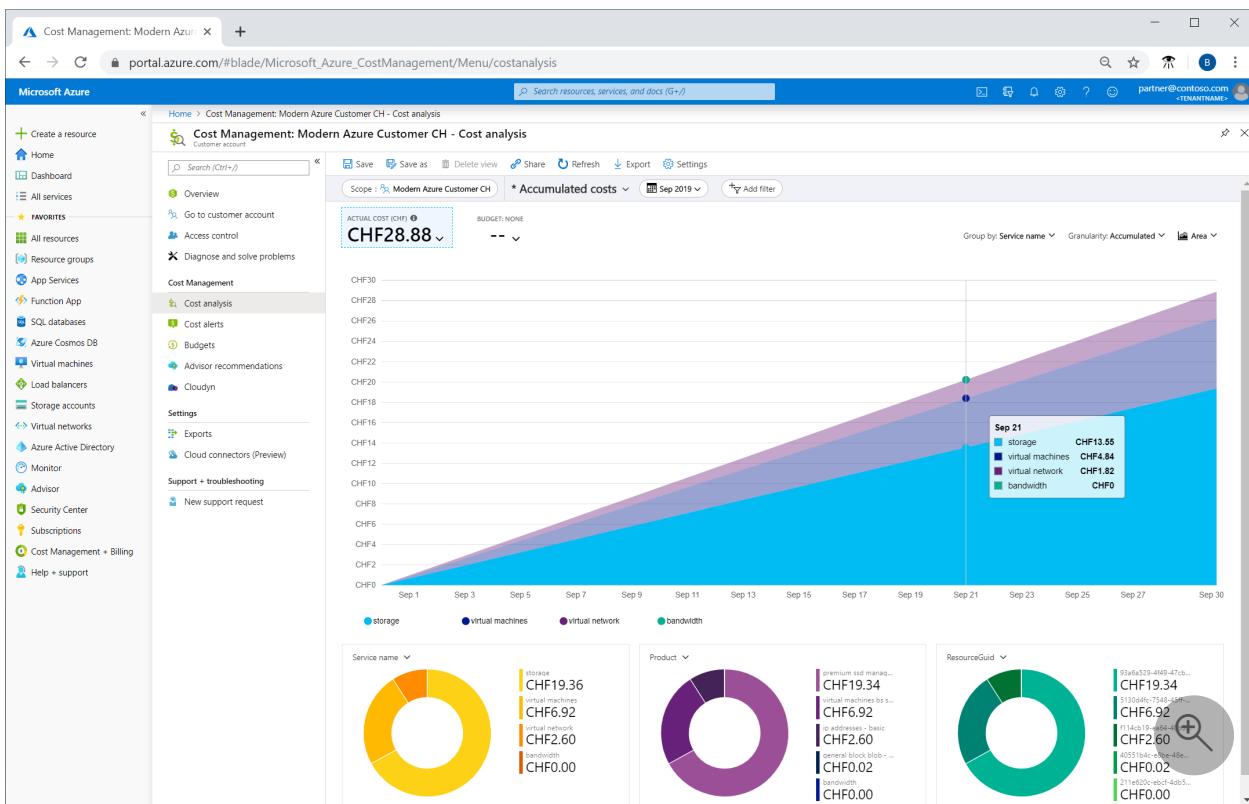
CSP partners use Cost Management to:

- Understand invoiced costs and associate the costs to the customer, subscriptions, resource groups, and services.
- Get an intuitive view of Azure costs in [cost analysis](#) with capabilities to analyze costs by customer, subscription, resource group, resource, meter, service, and many other dimensions.
- View resource costs that have Partner Earned Credit (PEC) applied in Cost Analysis.
- Set up notifications and automation using programmatic [budgets](#) and alerts when costs exceed budgets.
- Enable the Azure Resource Manager policy that provides customer access to Cost Management data. Customers can then view consumption cost data for their subscriptions using [pay-as-you-go rates](#).
- Export their cost and usage data to a storage blob with a pay-as-you-go subscription.

Here's an example showing costs for all customers.



Here's an example showing costs for a single customer.



All functionality available in Cost Management is also available with REST APIs. Use the APIs to automate cost management tasks.

Prerequisites

As a partner, Cost Management is natively available only for subscriptions that are on the Azure plan.

To enable Cost Management in the Azure portal, you must have confirmed customer acceptance of the Microsoft Customer Agreement (on behalf of the customer) and transitioned the customer to the Azure Plan. Only the costs for subscriptions that are transitioned to the Azure plan are available in Cost Management.

Cost Management requires read access to your billing account or subscription.

For more information about enabling and assigning access to Cost Management for a billing account, see [Assign users roles and permissions](#). The **Global admin** and **Admin agent** roles can manage costs for a billing account.

To access Cost Management at the subscription scope, any user with Azure RBAC access to a subscription can view costs at retail (pay-as-you-go) rates. However the [cost visibility policy for the customer tenant](#) must be enabled. To view a full list of supported account types, see [Understand Cost Management data](#).

When transferring existing billing agreements to a new partner, cost management capabilities are only available for the current billing relationship with the partner. Historical costs before the transfer to the new partner don't move to the new billing account. However, the cost history does remain with the original associated billing account.

How Cost Management uses scopes

Scopes are where you manage billing data, have roles specific to payments, view invoices, and conduct general account management. Billing and account roles are managed separately from scopes used for resource management, which use Azure RBAC. To clearly distinguish the intent of the separate scopes, including the access control differences, they are referred to as billing scopes and Azure RBAC scopes, respectively.

To understand billing scopes and Azure RBAC scopes and how cost management works with scopes, see [Understand and work with scopes](#).

Manage costs with partner tenant billing scopes

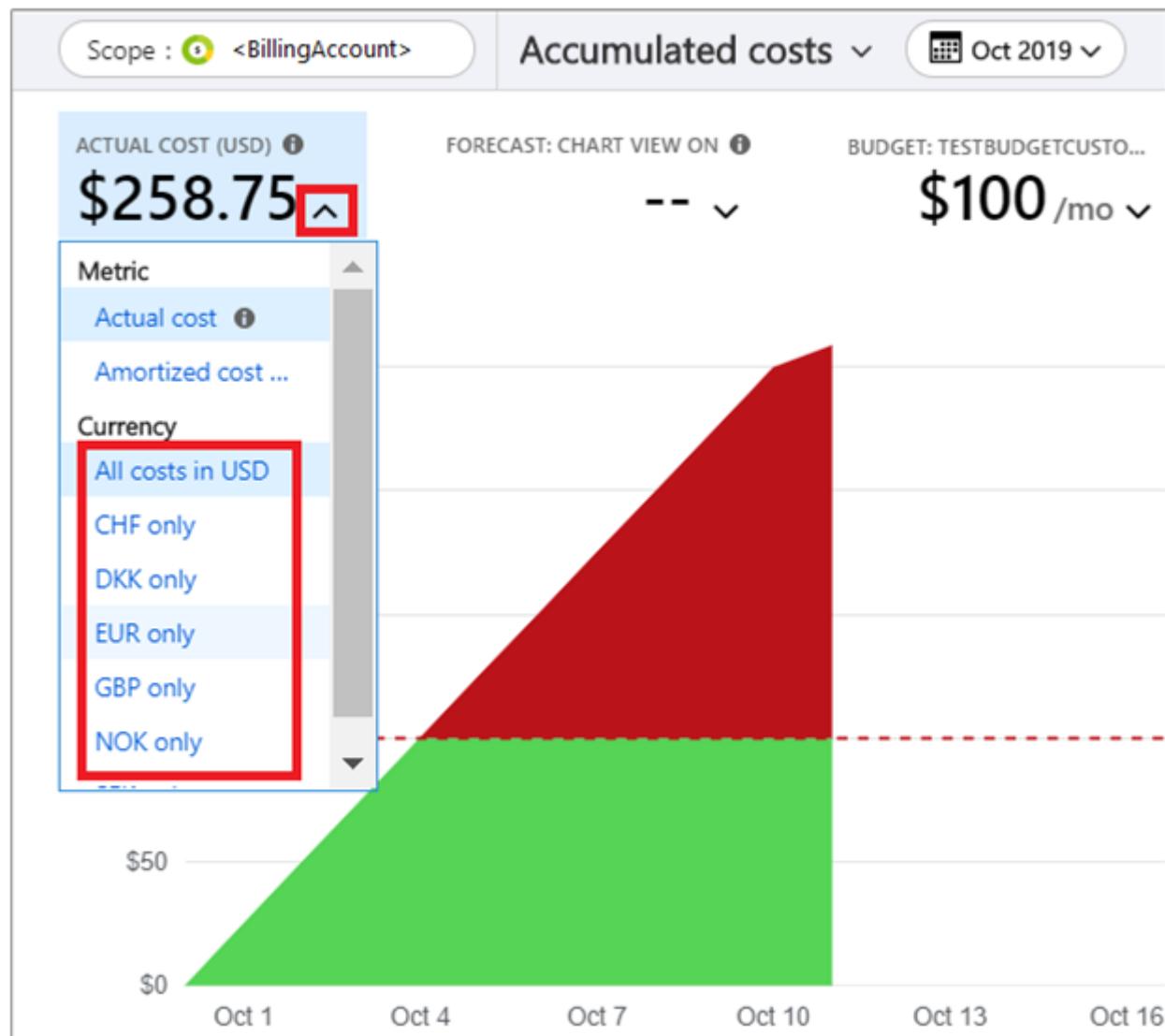
After you've onboarded your customers to a Microsoft Customer Agreement, the following *billing scopes* are available in your tenant. Use the scopes to manage costs in

Billing account scope

Use the billing account scope to view pre-tax costs across all your customers and billing profiles. Invoice costs are only shown for customer's consumption-based products on the Microsoft Customer Agreement. However, invoice costs are shown for purchased-based products for customers on both the Microsoft Customer Agreement and the CSP offer. Currently, the default currency to view costs in the scope is US dollars. Budgets set for the scope are also in USD.

Regardless of different billed currencies, partners use Billing account scope to set budgets and manage costs in USD across their customers, subscriptions, resources, and resource groups.

Partners also filter costs in a specific billing currency across customers in the cost analysis view. Select the **Actual cost** list to view costs in supported billing currencies.



Use the [amortized cost view](#) in billing scopes to view reserved instance amortized costs across a reservation term.

Billing profile scope

Use the billing profile scope to view pre-tax costs in the billing currency across all your customers for all products and subscriptions included in an invoice. You can filter costs in a billing profile for a specific invoice using the **InvoiceID** filter. The filter shows the consumption and product purchase costs for a specific invoice. You can also filter the costs for a specific customer on the invoice to see pre-tax costs.

After you onboard customers to a Microsoft Customer Agreement, you receive an invoice that includes all charges for all products (consumption, purchases, and entitlements) for these customers on the Microsoft Customer Agreement. When billed in the same currency, these invoices also include the charges for entitlement and purchased products such as SaaS, Azure Marketplace, and reservations for customers who are still in the classic CSP offer no on the Azure plan.

To help reconcile charges against the customer invoice, the billing profile scope enables you to see all costs that accrue for an invoice for your customers. Like the invoice, the scope shows costs for every customer in the new Microsoft Customer Agreement. The scope also shows every charge for customer entitlement products still in the current CSP offer.

The billing profile and billing account scopes are the only applicable scopes that show charges for entitlement and purchase-based products like Azure Marketplace and reservation purchases.

Billing profiles define the subscriptions that are included in an invoice. Billing profiles are the functional equivalent of an enterprise agreement enrollment. A billing profile is the scope where invoices are generated.

Currently, the billing currency is the default currency when viewing costs in the billing profile scope. Budgets set at the billing profile scope are in the billing currency.

Partners can use the scope to reconcile to invoices. And, they use the scope to set budgets in the billing currency for the following items:

- Specific filtered invoice
- Customer
- Subscription
- Resource group
- Resource

- Azure service
- Meter
- ResellerMPNID

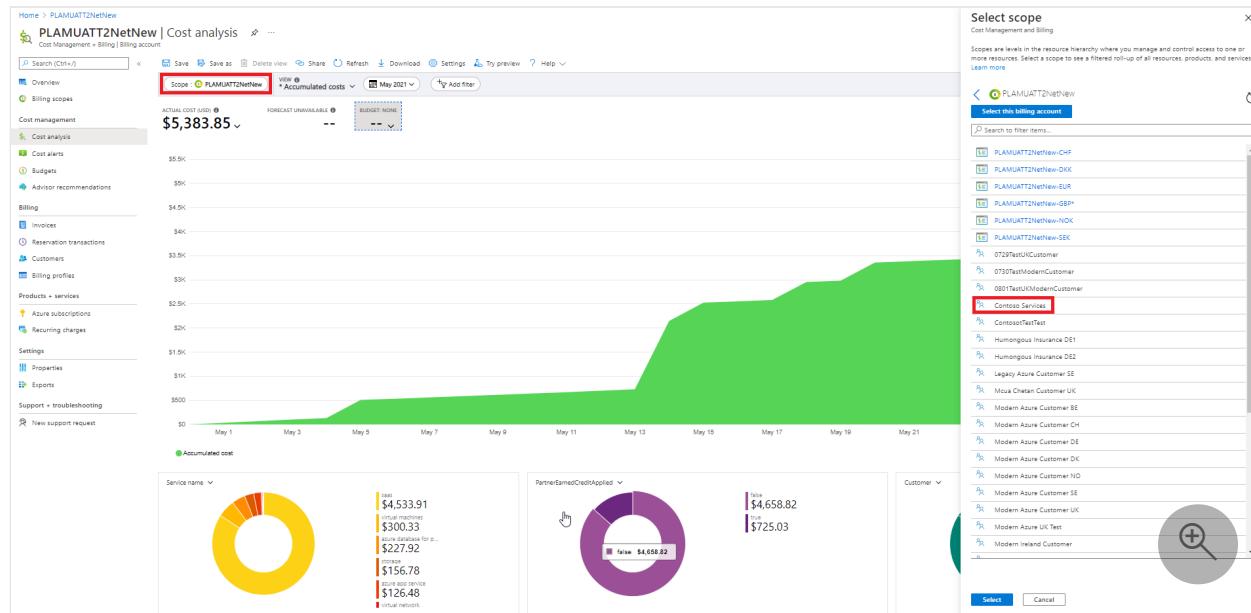
Customer scope

Partners use the scope to manage costs associated to customers that are onboarded to the Microsoft Customer Agreement. The scope allows partners to view pre-tax costs for a specific customer in a billing currency. You can also filter the pre-tax costs for a specific subscription, resource group, or resource.

The customer scope doesn't include customers who are on the current CSP offer. The scope only includes customers who have a Microsoft Customer Agreement.

Entitlement costs, not Azure usage, for current CSP offer customers are available at the billing account and billing profile scopes when you apply the customer filter. The budgets set at this scope are in the billing currency.

To view costs at the customer scope, in the partner tenant navigate to Cost analysis, select the scope picker and then select the specific customer in the list of scopes. Here's an example for the *Contoso Services* customer.



Partner access to billing scopes in Cost Management

Only the users with **Global admin** and **Admin agent** roles can manage and view costs for billing accounts, billing profiles, and customers directly in the partner's Azure tenant. For more information about partner center roles, see [Assign users roles and permissions](#).

Enable Cost Management for customer tenant subscriptions

Partners may enable access to Cost Management after customers are onboarded to a Microsoft Customer Agreement. Then partners can then enable a policy allowing customers to view their costs for Azure consumed services computed at pay-as-you-go retail rates. Costs are shown in the customer's billing currency for their consumed usage at Azure RBAC subscription and resource groups scopes.

When the policy for cost visibility is enabled by the partner, any user with Azure Resource Manager access to the subscription can manage and analyze costs at pay-as-you-go rates. Effectively, resellers and customers that have the appropriate Azure RBAC access to the Azure subscriptions can view cost.

Regardless of the policy, global admins and admin agents of the provider can view subscription costs if they have access to the subscription and resource group.

Enable the policy to view Azure usage charges

You need to be a member of the **admin agent** group to view and update the policy. Use the following information to enable the policy allowing customers to view Azure usage charges.

In the Azure portal, sign in to the *partner tenant* and select **Cost Management + Billing**. Select the relevant billing scope in the Billing Scope area, and then select **Customers**. The list of customers is associated with the billing account. *If you mistakenly sign in to the customer tenant, you won't see the **Customers** list.*

In the list of customers, select the customer that you want to allow to view costs.

The screenshot shows the Azure portal navigation bar on the left with various service icons. The 'Cost Management + Billing' icon is highlighted with a red box. The main content area shows a list of customers under the heading '<BillingAccount> - Customers'. A specific customer entry, '0729TestUKCustomer', is highlighted with a red box. The table displays columns for NAME, MONTH-TO-DATE C..., and LAST MONTH'S CHA... with corresponding values for each customer listed.

NAME	MONTH-TO-DATE C...	LAST MONTH'S CHA...
0729TestUKCustomer	GBP 130.14	0.00
0730TestModernCustomer	GBP 37.01	GBP 49.76
0801TestUKModernCustomer	0.00	0.00
Camila PAL testing	GBP 50.94	GBP 150.11
Contoso	0.00	0.00
ContosotTestTest	0.00	0.00
Johnny Modern Cust DE1	EUR 35.58	EUR 100.15
Johnny Modern Cust DE2	EUR 30.27	EUR 86.88

Under **Settings**, select **Policies**.

The current cost visibility policy is shown for **Azure Usage** charges associated to the subscriptions for the selected customer.

The screenshot shows the Azure Cost Management + Billing - Billing accounts interface. The URL is Home > Cost Management + Billing - Billing accounts > <BillingAccount> - Customers > 0729TestUKCustomer - Policies. The main title is "0729TestUKCustomer - Policies". On the left, there's a navigation menu with links like Overview, Cost management, Cost analysis, Budgets, Billing, Settings, Transfer requests, and Policies. The Policies link is highlighted with a red box. At the top right, there are "Save" and "Discard" buttons. A message states: "Users in 0729TestUKCustomer with access to an Azure subscription can view its charges at pay-as-you-go price." Below this is a "Yes" button (highlighted with a blue oval) and a "No" button.

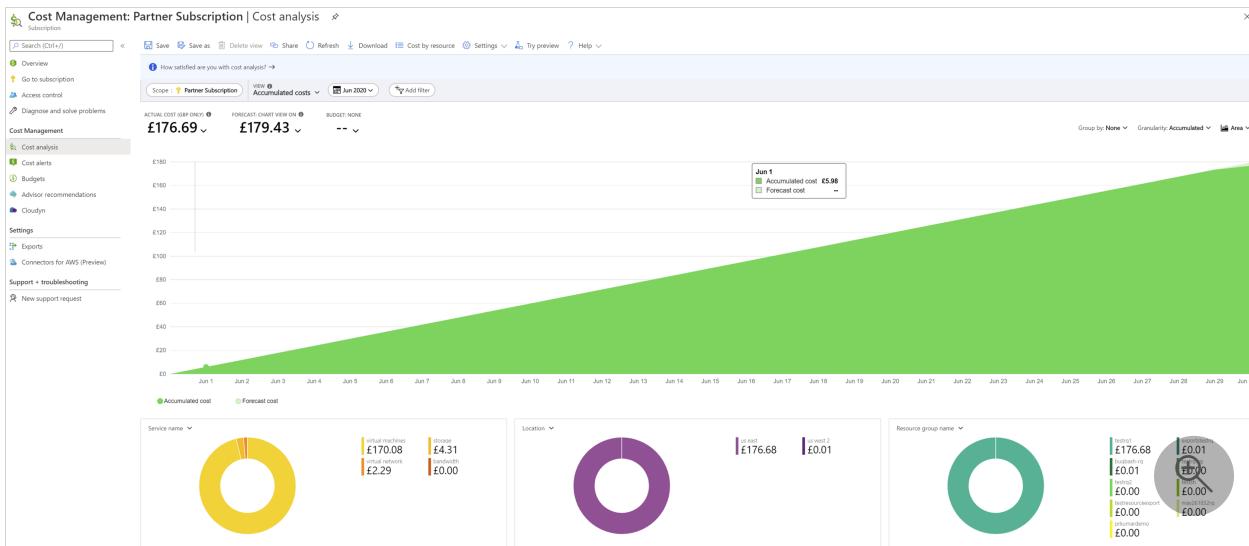
When the policy is set to **No**, Cost Management isn't available for subscription users associated to the customer. Unless enabled by a partner, the cost visibility policy is disabled by default for all subscription users.

When the cost policy is set to **Yes**, subscription users associated to the customer tenant can see usage charges at pay-as-you go rates.

When the cost visibility policy is enabled, all services that have subscription usage show costs at pay-as-you-go rates. Reservation usage appears with zero charges for actual and amortized costs. Purchases and entitlements are not associated to a specific subscription. So, purchases aren't displayed at the subscription scope. The global admin/admin agent of a direct partner or an indirect provider can also use the [Update Customer API](#) to set each customer's cost visibility policy at scale.

View subscription costs in the customer tenant

To view costs for a subscription, open **Cost Management** in the customer's Azure tenant. Select **Cost analysis** and then the required subscription to start reviewing costs. You can view consumption costs for each subscription individually in the customer tenant.



Cost analysis, budgets, and alerts are available for the subscription and resource group Azure RBAC scopes at pay-as-you-go rate-based costs.

Amortized views and actual costs for reserved instances in the Azure RBAC scopes show zero charges. Purchase costs for entitlements such as Reserved instances and Marketplace fees are only shown in billing scopes in the partner's tenant where the purchases were made.

The retail rates used to compute costs shown in the view are the same prices shown in the Azure Pricing Calculator for all customers. Costs shown don't include any discounts or credits that the partner may have like Partner Earned Credits, Tier Discounts, and Global Service discounts.

Analyze costs in cost analysis

Partners with access to billing scopes in the partner tenant can explore and analyze invoiced costs in cost analysis across customers for a specific customer or for an invoice. In cost analysis, you can also save views.

Azure RBAC users with access to the subscription in the customer tenant can also analyze retail costs for subscriptions in the customer tenant, save views, and export data to CSV and PNG files.

You can use filter and group by features in cost analysis to analyze costs by multiple fields. Partner-specific fields are shown in the next section.

Data fields

The following data fields are found in usage detail files and Cost Management APIs. Where available, Partner Center equivalent information is shown. For the following bold

fields, partners can use filter and group by features in cost analysis to analyze costs by multiple fields. Bold fields apply only to Microsoft Customer Agreements supported by partners.

Field name	Description	Partner Center equivalent
invoiceId	Invoice ID shown on the invoice for the specific transaction.	Invoice number where the transaction is shown.
previousInvoiceID	Reference to an original invoice there is a refund (negative cost). Populated only when there is a refund.	N/A
billingAccountName	Name of the billing account representing the partner. It accrues all costs across the customers who have onboarded to a Microsoft customer agreement and the CSP customers that have made entitlement purchases like SaaS, Azure Marketplace, and reservations.	N/A
billingAccountId	Identifier for the billing account representing the partner.	MCAPI Partner Commerce Root ID. Used in a request, but not included in a response.
billingProfileID	Identifier for the billing profile that groups costs across invoices in a single billing currency across the customers who have onboarded to a Microsoft customer agreement and the CSP customers that have made entitlement purchases like SaaS, Azure Marketplace, and reservations.	MCAPI Partner Billing Group ID. Used in a request, but not included in a response.

Field name	Description	Partner Center equivalent
billingProfileName	Name of the billing profile that groups costs across invoices in a single billing currency across the customers who have onboarded to a Microsoft customer agreement and the CSP customers that have made entitlement purchases like SaaS, Azure Marketplace, and reservations.	N/A
invoiceSectionName	Name of the project that is being charged in the invoice. Not applicable for Microsoft Customer Agreements onboarded by partners.	N/A
invoiceSectionID	Identifier of the project that is being charged in the invoice. Not applicable for Microsoft Customer Agreements onboarded by partners.	N/A
CustomerTenantID	Identifier of the Azure Active Directory tenant of the customer's subscription.	Customer's organizational ID - the customer's Azure Active Directory TenantID.
CustomerName	Name of the Azure Active Directory tenant for the customer's subscription.	Customer's organization name, as shown in the Partner Center. Important for reconciling the invoice with your system information.
CustomerTenantDomainName	Domain name for the Azure Active Directory tenant of the customer's subscription.	Customer Azure Active Directory tenant domain.
PartnerTenantID	Identifier for the partner's Azure Active Directory tenant.	Partner Azure Active Directory Tenant ID called as Partner ID, in GUID format.
PartnerName	Name of the partner Azure Active Directory tenant.	Partner name.
ResellerMPNID	ID for the reseller associated with the subscription.	ID of the reseller on record for the subscription. Not available for current activity.

Field name	Description	Partner Center equivalent
costCenter	Cost center associated to the subscription.	N/A
billingPeriodStartDate	Billing period start date, as shown on the invoice.	N/A
billingPeriodEndDate	Billing period end date, as shown on the invoice.	N/A
servicePeriodStartDate	Start date for the rating period when the service usage was rated for charges. The prices for Azure services are determined for the rating period.	ChargeStartDate in Partner Center. Billing cycle start date, except when presenting dates of previously uncharged latent usage data from a previous billing cycle. The time is always the beginning of the day, 0:00.
servicePeriodEndDate	End date for the period when the service usage was rated for charges. The prices for Azure services are determined based on the rating period.	N/A
date	For Azure consumption data, it shows date of usage as rated. For reserved instance, it shows the purchased date. For recurring charges and one-time charges such as Marketplace and support, it shows the purchase date.	N/A
productID	Identifier for the product that has accrued charges by consumption or purchase. It is the concatenated key of productID and SKuID, as shown in the Partner Center.	The ID of the product.
product	Name of the product that has accrued charges by consumption or purchase, as shown on the invoice.	The product name in the catalog.

Field name	Description	Partner Center equivalent
serviceFamily	Shows the service family for the product purchased or charged. For example, Storage or Compute.	N/A
productOrderID	The identifier of the asset or Azure plan name that the subscription belongs to. For example, Azure Plan.	CommerceSubscriptionID
productOrderName	The name of the Azure plan that the subscription belongs to. For example, Azure Plan.	N/A
consumedService	Consumed service (legacy taxonomy) as used in legacy EA usage details.	Service shown in the Partner Center. For example, Microsoft.Storage, Microsoft.Compute, and microsoft.operationalinsights.
meterID	Metered identifier for measured consumption.	The ID of the used meter.
meterName	Identifies the name of the meter for measured consumption.	The name of the consumed meter.
meterCategory	Identifies the top-level service for usage.	The top-level service for the usage.
meterSubCategory	Defines the type or subcategory of Azure service that can affect the rate.	The type of Azure service that can affect the rate.
meterRegion	Identifies the location of the datacenter for certain services that are priced based on datacenter location.	The regional location of a data center for services, where applicable and populated.
subscription ID	Unique Microsoft generated identifier for the Azure subscription.	EntitlementID
subscriptionName	Name of the Azure subscription.	N/A

Field name	Description	Partner Center equivalent
Term	Displays the term for the validity of the offer. For example, reserved instances show 12 months of a yearly term of the reserved instance. For one-time purchases or recurring purchases, the term displays one month for SaaS, Azure Marketplace, and support. Not applicable for Azure consumption.	N/A
provider	Identifier for product and line of business. Break down costs by the provider type: Azure, Microsoft 365, Dynamics 365, AWS, and so on.	N/A
publisherType (firstParty, thirdPartyReseller, thirdPartyAgency)	Type of publisher that identifies the publisher as first party, third-party reseller, or third-party agency.	N/A
partNumber	Part number for the unused reserved instance and Azure Marketplace services.	N/A
publisherName	Name of the publisher of the service including Microsoft or third-party publishers.	The name of the product's publisher.
reservationId	Identifier for the reserved instance purchase.	N/A
reservationName	Name of the reserved instance.	N/A
reservationOrderId	OrderID for the reserved instance.	N/A
frequency	Payment frequency for a reserved instance.	N/A
resourceGroup	Name of the Azure resource group used for lifecycle resource management.	Name of the resource group.
instanceID (or) ResourceID	Identifier of the resource instance.	Shown as a ResourceURI that includes complete resource properties.

Field name	Description	Partner Center equivalent
resourceLocation	Name of the resource location.	The location of the resource.
Location	Normalized location of the resource.	N/A
effectivePrice	The effective unit price of the service, in pricing currency. Unique for a product, service family, meter, and offer. Used with pricing in the price sheet for the billing account. When there is tiered pricing or an included quantity, it shows the blended price for consumption.	The unit price after adjustments are made.
Quantity	Measured quantity purchased or consumed. The amount of the meter used during the billing period.	Number of units. Ensure it matches the information in your billing system during reconciliation.
unitOfMeasure	Identifies the unit that the service is charged in. For example, GB and hours.	Identifies the unit that the service is charged in. For example, GB, hours, and 10,000 s.
pricingCurrency	The currency defining the unit price.	The currency in the price list.
billingCurrency	The currency defining the billed cost.	The currency defined as the billed currency on the invoice.
chargeType	Defines the type of charge that the cost represents in Cost Management like purchase and refund.	The type of charge or adjustment. Not available for current activity.
costInBillingCurrency	ExtendedCost or blended cost before tax in the billed currency.	N/A
costInPricingCurrency	ExtendedCost or blended cost before tax in pricing currency to correlate with prices.	N/A
costInUSD	Estimated ExtendedCost or blended cost before tax in USD.	N/A

Field name	Description	Partner Center equivalent
paygCostInBillingCurrency	Shows costs if pricing is in retail prices. Shows pay-as-you-go prices in the billing currency. Available only at Azure RBAC scopes.	N/A
paygCostInUSD	Shows costs if pricing is in retail prices. Shows pay-as-you-go prices in USD. Available only at Azure RBAC scopes.	N/A
exchangeRate	Exchange rate used to convert from the pricing currency to the billing currency.	Referred to as PCToBCExchangeRate in the Partner Center. The pricing currency to billing currency exchange rate.
exchangeRateDate	The date for the exchange rate that's used to convert from the pricing currency to the billing currency.	Referred to as PCToBCExchangeRateDat in the Partner Center. The pricing currency to billing currency exchange rate date.
isAzureCreditEligible	Indicates whether the cost is eligible for payment by Azure credits.	N/A
serviceInfo1	Legacy field that captures optional service-specific metadata.	Internal Azure service metadata.
serviceInfo2	Legacy field that captures optional service-specific metadata.	Service information. For example, an image type for a virtual machine and ISP name for ExpressRoute.
additionalInfo	Service-specific metadata. For example, an image type for a virtual machine.	Any additional information not covered in other columns. The service-specific metadata. For example, an image type for a virtual machine.

Field name	Description	Partner Center equivalent
tags	Tag that you assign to the meter. Use tags to group billing records. For example, you can use tags to distribute costs by the department that uses the meter.	Tags added by the customer.
partnerEarnedCreditRate	Rate of discount applied if there is a partner earned credit (PEC) based on partner admin link access.	The rate of partner earned credit (PEC). For example, 0% or 15%.
partnerEarnedCreditApplied	Indicates whether the partner earned credit has been applied.	N/A

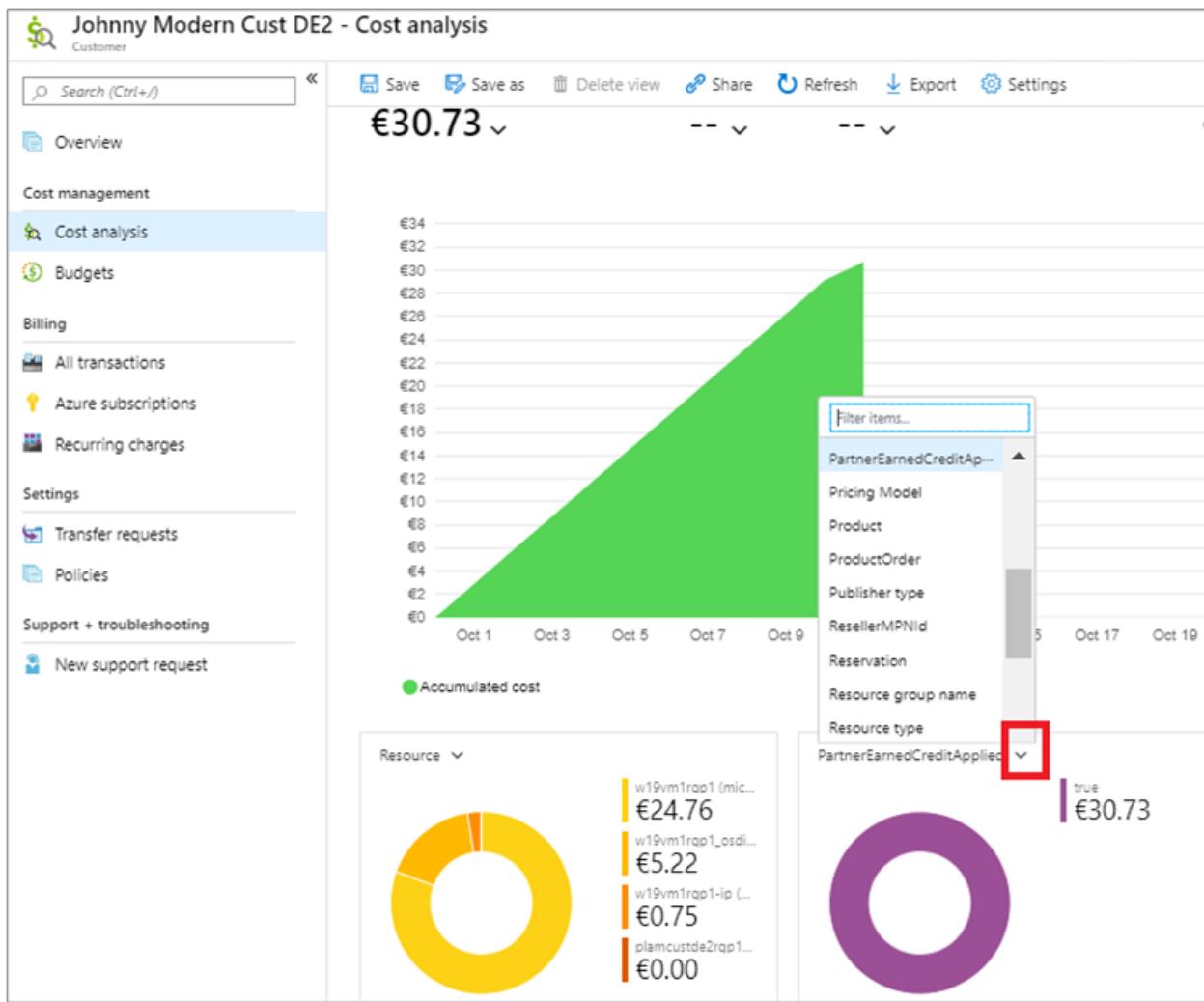
View Partner Earned Credit (PEC) resource costs

In Cost Management, partners can use cost analysis to view costs that received the PEC benefits.

In the Azure portal, sign in to the partner tenant and select **Cost Management + Billing**. Under **Cost Management**, select **Cost analysis**.

The Cost analysis view shows costs of the billing account for the partner. Select the **Scope** as needed for the partner, a specific customer, or a billing profile to reconcile invoices.

In a donut chart, select the drop-down list and select **PartnerEarnedCreditApplied** to drill into PEC costs.

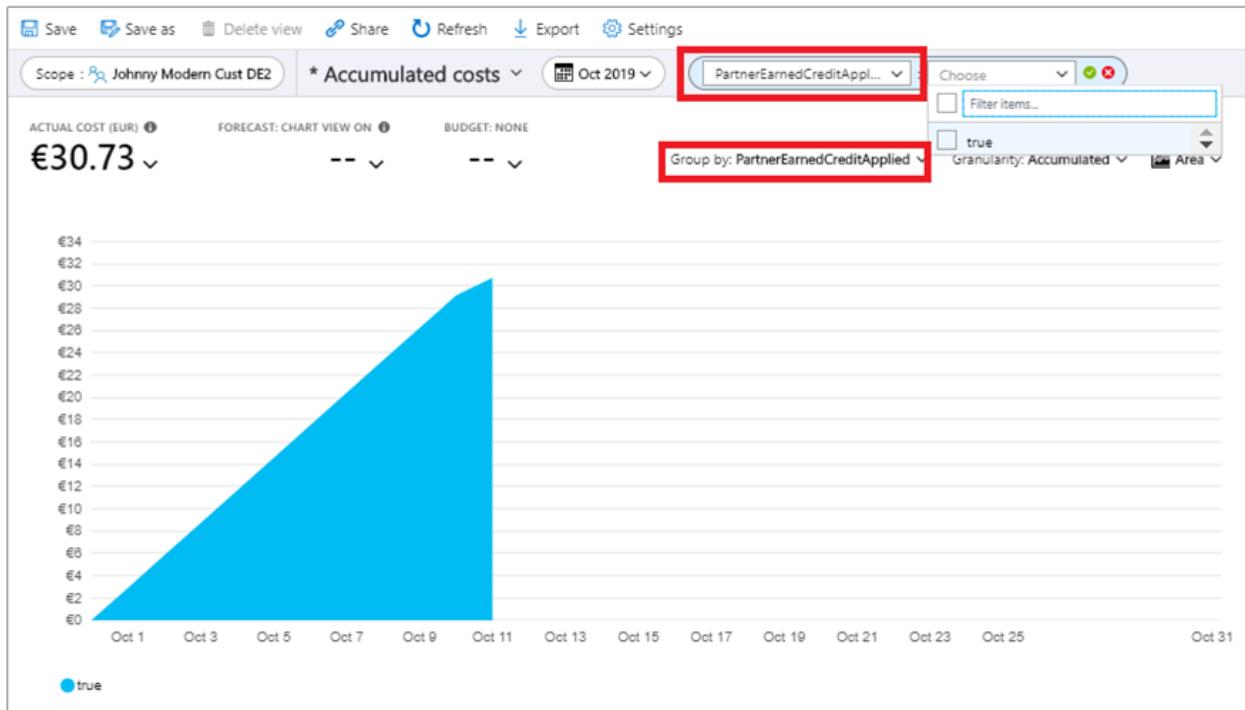


When the **PartnerEarnedCreditApplied** property is *True*, the associated cost has the benefit of the partner earned admin access.

When the **PartnerEarnedCreditApplied** property is *False*, the associated cost hasn't met the required eligibility for the credit. Or, the service purchased isn't eligible for partner earned credit.

Service usage data normally takes 8-24 hours to appear in Cost Management. For more information, see [Cost and usage data updates and retention](#). PEC credits appear within 48 hours from time of access in Cost Management.

You can also group and filter by the **PartnerEarnedCreditApplied** property using the **Group by** options. Use the options to examine costs that do and don't have PEC.



Export cost data to Azure Storage

Partners with access to billing scopes in a partner tenant can export their cost and usage data to an Azure Storage blob. The blob must be on a subscription in the partner tenant that's not a [shared service subscription](#) or a customer's subscription. To enable cost data export, we recommend that you set up an independent pay-as-you-go subscription in the partner tenant to host the exported cost data. The export storage account is created on the Azure Storage blob hosted in the pay-as-you-go subscription. Based on the scope where the partner creates the export, the associated data is exported to the storage account automatically on a recurring basis.

Users with Azure RBAC access to the subscription can also export the cost data to an Azure storage blob hosted in any subscription in the customer tenant.

Create an export in a partner tenant or customer tenant

In the Azure portal, sign in to the partner tenant or customer tenant and select **Cost Management + Billing**. Select an appropriate scope, for example a Microsoft Partner Agreement billing account, and then select **Cost Analysis**. When the page loads, select **Export**. Select **View all exports** under Schedule Export.

The screenshot shows the Azure Cost Management interface for the Contoso (Demo) tenant. The main area displays cost analysis metrics: Actual Cost (USD) \$47,866.85, Forecast: Chart View on \$99,409.09, and Budget: --. The sidebar on the left includes links for Overview, Go to billing account, Access control, Diagnose and solve problems, Cost Management, Cost analysis (which is selected), and Cost alerts. The top navigation bar has Save, Save as, Delete view, Share, Refresh, Export, Settings, and Try preview buttons. The Export button is highlighted with a red box.

Next, select **Add** and type the name and select an export type. Select the **Storage** tab and enter required information.

The screenshot shows the 'Exports' blade in the Azure portal. On the left, there's a list of existing exports with a red box around the '+ Add' button. On the right, the 'New export' configuration pane is open. It has three tabs: Basics, Storage (which is selected and highlighted with a red box), and Review. Under Basics, there are two radio buttons: 'Use existing' (selected) and 'Create new'. Under Storage, fields are filled with 'Subscription': 'Cost Management PM', 'Storage account': 'acmblob01', 'Container': 'example-container', and 'Directory': 'exports'. At the bottom right of the pane is a 'Create' button.

When you create an export in the partner tenant, select the pay-as-you-go subscription in the partner tenant. Create an Azure Storage account using that subscription.

For Azure RBAC users in the customer tenant, select a subscription in the customer tenant. Create an Azure Storage account using the subscription.

Review the content and then select **Create** to schedule an export.

To verify data in the export list, select the storage account name. On the storage account page, select **Containers** and then select the container. Navigate to the corresponding folder and select the CSV file. Select **Download** to get the CSV file and open it. The exported data exported resembles cost data similar to usage details from the Azure portal.

A	B	C	D	E	F	G	H	I
Invoiceld	PreviousInv	BillingAcco	BillingAcco	BillingProfi	BillingProfi	InvoiceSect	InvoiceSect	PartnerTen
G000493998		aff095f4-f2	PLAMUATT	5XQV-V4Uf	PLAMUATT	WBEJ-VB7F-PJA-AJ4D-`0e195b37-		
G000493998		aff095f4-f2	PLAMUATT	5XQV-V4Uf	PLAMUATT	WBEJ-VB7F-PJA-AJ4D-`0e195b37-		
G000493998		aff095f4-f2	PLAMUATT	5XQV-V4Uf	PLAMUATT	WBEJ-VB7F-PJA-AJ4D-`0e195b37-		

Cost Management REST APIs

Partners and their customers can use Cost Management APIs for common tasks. For more information, see [Automation for partners](#).

Next steps

- [Start analyzing costs](#) in Cost Management
- [Create and manage budgets](#) in Cost Management

Azure security baseline for Cost Management

Article • 10/12/2022

This security baseline applies guidance from the [Microsoft cloud security benchmark version 1.0](#) to Cost Management. The Microsoft cloud security benchmark provides recommendations on how you can secure your cloud solutions on Azure. The content is grouped by the security controls defined by the Microsoft cloud security benchmark and the related guidance applicable to Cost Management.

You can monitor this security baseline and its recommendations using Microsoft Defender for Cloud. Azure Policy definitions will be listed in the Regulatory Compliance section of the Microsoft Defender for Cloud dashboard.

When a feature has relevant Azure Policy Definitions, they are listed in this baseline to help you measure compliance to the Microsoft cloud security benchmark controls and recommendations. Some recommendations may require a paid Microsoft Defender plan to enable certain security scenarios.

ⓘ Note

Features not applicable to Cost Management have been excluded. To see how Cost Management completely maps to the Microsoft cloud security benchmark, see the [full Cost Management security baseline mapping file ↗](#).

Security profile

The security profile summarizes high-impact behaviors of Cost Management, which may result in increased security considerations.

Service Behavior Attribute	Value
Product Category	MGMT/Governance
Customer can access HOST / OS	No Access
Service can be deployed into customer's virtual network	False
Stores customer content at rest	True

Network security

For more information, see the [Microsoft cloud security benchmark: Network security](#).

NS-2: Secure cloud services with network controls

Features

Disable Public Network Access

Description: Service supports disabling public network access either through using service-level IP ACL filtering rule (not NSG or Azure Firewall) or using a 'Disable Public Network Access' toggle switch. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
False	Not Applicable	Not Applicable

Configuration Guidance: This feature is not supported to secure this service.

Identity management

For more information, see the [Microsoft cloud security benchmark: Identity management](#).

IM-1: Use centralized identity and authentication system

Features

Azure AD Authentication Required for Data Plane Access

Description: Service supports using Azure AD authentication for data plane access. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	True	Microsoft

Feature notes: Azure Cost Management is integrated with Azure Active Directory (Azure AD) as the services operates within the Azure Portal.

Configuration Guidance: No additional configurations are required as this is enabled on a default deployment.

Reference: [Manage access to billing information for Azure](#)

Local Authentication Methods for Data Plane Access

Description: Local authentications methods supported for data plane access, such as a local username and password. [Learn more.](#)

Supported	Enabled By Default	Configuration Responsibility
False	Not Applicable	Not Applicable

Configuration Guidance: This feature is not supported to secure this service.

IM-3: Manage application identities securely and automatically

Features

Managed Identities

Description: Data plane actions support authentication using managed identities. [Learn more.](#)

Supported	Enabled By Default	Configuration Responsibility
False	Not Applicable	Not Applicable

Configuration Guidance: This feature is not supported to secure this service.

Service Principals

Description: Data plane supports authentication using service principals. [Learn more.](#)

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: In order to use Azure Cost Management APIs, you need to properly assign permissions to an Azure service principal. From there you can use the service

principal identity to call the APIs.

Configuration Guidance: For services that don't support managed identities, use Azure Active Directory (Azure AD) to create a service principal with restricted permissions at the resource level. Configure service principals with certificate credentials and fall back to client secrets for authentication.

After you create a service principal to programmatically call the Azure Resource Manager APIs, you need to assign it the proper permissions to authorize against and execute requests in Azure Resource Manager.

Reference: [Assign permissions to Cost Management APIs](#)

IM-7: Restrict resource access based on conditions

Features

Conditional Access for Data Plane

Description: Data plane access can be controlled using Azure AD Conditional Access Policies. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Feature notes: Azure Cost Management is integrated with Azure Active Directory (Azure AD). Conditional access policies may be configured by the customer to authenticate via Azure AD.

Configuration Guidance: Define the applicable conditions and criteria for Azure Active Directory (Azure AD) conditional access in the workload. Consider common use cases such as blocking or granting access from specific locations, blocking risky sign-in behavior, or requiring organization-managed devices for specific applications.

Reference: [What is Conditional Access?](#)

IM-8: Restrict the exposure of credential and secrets

Features

Service Credential and Secrets Support Integration and Storage in Azure Key Vault

Description: Data plane supports native use of Azure Key Vault for credential and secrets store. [Learn more.](#)

Supported	Enabled By Default	Configuration Responsibility
False	Not Applicable	Not Applicable

Configuration Guidance: This feature is not supported to secure this service.

Privileged access

For more information, see the [Microsoft cloud security benchmark: Privileged access](#).

PA-1: Separate and limit highly privileged/administrative users

Features

Local Admin Accounts

Description: Service has the concept of a local administrative account. [Learn more.](#)

Supported	Enabled By Default	Configuration Responsibility
False	Not Applicable	Not Applicable

Configuration Guidance: This feature is not supported to secure this service.

PA-7: Follow just enough administration (least privilege) principle

Features

Azure RBAC for Data Plane

Description: Azure Role-Based Access Control (Azure RBAC) can be used to manage access to service's data plane actions. [Learn more.](#)

Supported	Enabled By Default	Configuration Responsibility
True	False	Customer

Configuration Guidance: Use Azure role-based access control (Azure RBAC) to manage Azure resource access through built-in role assignments. After an Account administrator has assigned the appropriate roles to other users, they must turn on access to download invoices in the Azure portal.

Reference: [Manage access to billing information for Azure](#)

Data protection

For more information, see the [Microsoft cloud security benchmark: Data protection](#).

DP-3: Encrypt sensitive data in transit

Features

Data in Transit Encryption

Description: Service supports data in-transit encryption for data plane. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	True	Microsoft

Feature notes: By default, data in transit is encrypted by Microsoft.

Configuration Guidance: No additional configurations are required as this is enabled on a default deployment.

Reference: [Double encryption](#)

DP-4: Enable data at rest encryption by default

Features

Data at Rest Encryption Using Platform Keys

Description: Data at-rest encryption using platform keys is supported, any customer content at rest is encrypted with these Microsoft managed keys. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
True	True	Microsoft

Feature notes: By default, data at rest is automatically encrypted by Microsoft using platform-managed encryption keys.

Configuration Guidance: No additional configurations are required as this is enabled on a default deployment.

Reference: [Double encryption](#)

DP-5: Use customer-managed key option in data at rest encryption when required

Features

Data at Rest Encryption Using CMK

Description: Data at-rest encryption using customer-managed keys is supported for customer content stored by the service. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility
False	Not Applicable	Not Applicable

Configuration Guidance: This feature is not supported to secure this service.

DP-6: Use a secure key management process

Features

Key Management in Azure Key Vault

Description: The service supports Azure Key Vault integration for any customer keys, secrets, or certificates. [Learn more](#).

Supported	Enabled By Default	Configuration Responsibility

Supported	Enabled By Default	Configuration Responsibility
False	Not Applicable	Not Applicable

Configuration Guidance: This feature is not supported to secure this service.

Asset management

For more information, see the [Microsoft cloud security benchmark: Asset management](#).

AM-2: Use only approved services

Features

Azure Policy Support

Description: Service configurations can be monitored and enforced via Azure Policy.

[Learn more.](#)

Supported	Enabled By Default	Configuration Responsibility
False	Not Applicable	Not Applicable

Configuration Guidance: This feature is not supported to secure this service.

Backup and recovery

For more information, see the [Microsoft cloud security benchmark: Backup and recovery](#).

BR-1: Ensure regular automated backups

Features

Azure Backup

Description: The service can be backed up by the Azure Backup service. [Learn more.](#)

Supported	Enabled By Default	Configuration Responsibility
False	Not Applicable	Not Applicable

Configuration Guidance: This feature is not supported to secure this service.

Next steps

- See the [Microsoft cloud security benchmark overview](#)
- Learn more about [Azure security baselines](#)

Troubleshoot common Cost Management errors

Article • 12/14/2022

This article describes common Cost Management errors and provides information about solutions. When you use Cost Management in the Azure portal and encounter an error that you don't understand or can't resolve, find the error code below. Then try to use the mitigation information or the more information link to resolve the problem.

Here's a list of common error codes with mitigation information.

If the information provided doesn't help you, [Create a support request](#).

400

Error message `400`.

Mitigation

If you're using the [BillingPeriods API](#), confirm that you're using a classic pay-as-you-go or EA subscription. The BillingPeriods API doesn't support Microsoft Customer Agreement subscriptions.

Confirm that you're using a supported scope for the specific feature or subscription offer type.

There are many feature-specific errors that use the `400` error code. Refer to the error message and API documentation for specific details. For general information, see [Cost Management APIs](#).

More information

For more information about billing periods when transitioning to a Microsoft Customer Agreement, see [Billing period](#).

401

Error message `401`.

Mitigation

For an Enterprise Agreement, confirm that the view charges options (Account Owner or Department Administrator) have been enabled.

For a Microsoft Customer Agreement, confirm that the billing account owner has assigned you to a role that can view charges.

See [AuthorizationFailed](#).

More information

For more information about enterprise agreements, see [Troubleshoot enterprise cost views](#).

For more information about Microsoft Customer Agreements, see [Understand Microsoft Customer Agreement administrative roles in Azure](#).

404

Error message [404](#).

Mitigation

Confirm that you're using a supported scope for the specific feature or supported subscription offer type.

Also, see [NotFound](#).

500

Error message [500](#).

Mitigation

This message is an internal error. Wait an hour and try again.

Also, see [GatewayTimeout](#).

503

Error message [503](#).

Mitigation

This message is an internal error. Wait an hour and try again.

When creating or updating exports, you might view the error when the Microsoft.CostManagementExports resource provider is being registered for your subscription. Resource provider registration is quick, but you may need to wait up to five minutes. If you still see the error for more than 10 minutes, [create a support request](#).

Also, see [GatewayTimeout](#).

AccountCostDisabled

Error message `AccountCostDisabled`.

Mitigation

The message indicates that the Enterprise Agreement administrator hasn't enabled Cost Management (view charges) for account owners and subscription users. Contact your administrator.

More information

For more information, see [Troubleshoot Azure enterprise cost views](#).

AuthorizationFailed

Error message `AuthorizationFailed`.

Mitigation

Confirm that you have access to the specified scope or object. For example, budget or export.

More information

For more information, see [Assign access to Cost Management data](#)

BadRequest

Error message `BadRequest`.

Mitigation

When using the Query or Forecast APIs to retrieve cost data, validate the query body.

When using portal experiences and you see the `object ID cannot be null` error, try refreshing your view.

When using Power BI to pull reservation usage data for more than 3 months, you will need to break down the call into 3-month chunks.

Also, see [SubscriptionTypeNotSupported](#).

More information

For more information about the Query - Usage API body examples, see [Query - Usage](#).

For more information about the Forecast - Usage API body examples, see [Forecast - Usage](#).

For more information about chunking reservation usage calls in Power BI, see [Power BI considerations and limitations](#).

BillingAccessDenied

Error message `BillingAccessDenied`.

Mitigation

See [AuthorizationFailed](#).

DepartmentCostDisabled

Error message `DepartmentCostDisabled`.

Mitigation

The message indicates that the Enterprise Agreement administrator hasn't enabled Cost Management (DA view charges) for department admins. Contact your EA administrator.

More information

For more information about troubleshooting disabled costs, see [Troubleshoot Azure enterprise cost views](#).

DisallowedOperation

Error message `DisallowedOperation`.

Mitigation

The message indicates that the subscription doesn't have any charges. The type of subscription that you're using isn't allowed to incur charges. Because the subscription can't have any billed charges, it isn't supported by Cost Management.

FailedDependency

Error message `FailedDependency`.

Mitigation

When you're using the Forecast API, the error indicates that there's either not enough data to generate an accurate forecast. Or, there are multiple currencies that can't be merged.

If you have multiple currencies, filter down to charges that only have one currency or request an aggregation of `CostUSD` instead of `Cost` to get a forecast normalized to USD.

If there's not enough historical data, wait one week since you first had charges on the scope to see a forecast.

More information

For more information about the API, see [Forecast - Usage](#).

GatewayTimeout

Error message `GatewayTimeout`.

Mitigation

The message is an internal error. Wait an hour and try again.

When querying for cost data using the Query, Forecast, or Publish APIs, consider simplifying your query with less group by columns or using a lower-level scope. Avoid using large management groups with more than 50 subscriptions.

IndirectCostDisabled

Error message `IndirectCostDisabled`.

Mitigation

The message indicates that your partner hasn't published pricing for the Enterprise Agreement enrollment, which is required to use Cost Management. Contact your partner.

More information

For more information, see [Troubleshoot Azure enterprise cost views](#).

InvalidAuthenticationTokenTenant

Error message `InvalidAuthenticationTokenTenant`.

Mitigation

The subscription you're accessing might have been moved to a different directory.

When using the Azure portal, you might have used a link or saved reference, like a dashboard tile, before the subscription was moved.

Switch to the correct directory that was mentioned in the error message and try again. Don't forget to remove any old references and update any links.

InvalidGatewayHost

Error message `InvalidGatewayHost`.

Mitigation

The message is an internal error. Try again in five minutes. If the error continues, [create a support request](#).

InvalidScheduledActionEmailRecipients

Error message `InvalidScheduledActionEmailRecipients`.

Mitigation

The message indicates that the scheduled action/email for an alert that you're creating or updating doesn't have any email recipients. When using the Azure portal, press ENTER after specifying an email address to ensure it's saved in the form.

InvalidView

Error message `InvalidView`.

Mitigation

The message indicates that the view specified when creating or updating an alert with the ScheduledActions API isn't valid.

When configuring anomaly alerts, make sure you use a kind value of `InsightAlert`.

MissingSubscription

Error message `MissingSubscription`.

Mitigation

The message indicates that the HTTP request didn't include a valid scope.

If using the Azure portal, [create a support request](#). The error is likely caused by an internal problem.

NotFound

Error message `NotFound`.

Mitigation

If using a subscription or resource group, see [SubscriptionNotFound](#).

If using a management group, see [SubscriptionTypeNotSupportedException](#).

If using Cost Management in the Azure portal, try refreshing the page. The error may be caused by an old reference to a deleted object within the system, like a budget or connector.

For any other cases, validate the scope or resource ID.

More information

For more information, see [Assign access to Cost Management data](#).

RBACAccessDenied

Indicates that the current user/account does not have adequate Role-Based Access Control (RBAC) permission to perform the action.

Mitigation

If creating a budget that references an action group (`contactGroups` in the request body), make sure the user/account executing the PUT request has both Cost Management Contributor (or `Microsoft.Consumption/budgets/write`) access as well as Monitoring Reader (or `Microsoft.Insights/actionGroups/read`) access.

For additional mitigation steps, see [AuthorizationFailed](#).

ReadOnlyDisabledSubscription

Error message `ReadOnlyDisabledSubscription`.

Mitigation

The subscription is disabled. You can't create or update Cost Management objects, like budgets and views, for a disabled subscription.

More information

For more information, see [Reactivate a disabled Azure subscription](#).

ResourceGroupNotFound

Mitigation

The error indicates that a resource group doesn't exist. The resource group might be moved or deleted.

If using the Azure portal, you might see the error when creating budgets or exports. The error is expected and you can ignore it.

ResourceRequestsThrottled

Error message `ResourceRequestsThrottled`.

Mitigation

The error is caused by excessive use within a short timeframe. Wait five minutes and try again.

More information

For more information, see [Data latency and rate limits](#).

ServerTimeout

Error message `ServerTimeout`.

Mitigation

For mitigation information, see [GatewayTimeout](#).

SubscriptionNotFound

Error message `SubscriptionNotFound`.

Mitigation

- Validate that the subscription ID is correct.
- Confirm that you have a supported subscription type.

If using Cost Management for a newly created subscription, wait 48 hours and try again.

More information

Supported subscription types are shown at [Understand Cost Management data](#).

SubscriptionTypeNotSupported

Error message `SubscriptionTypeNotSupported`.

Mitigation

If using a management group, verify that all subscriptions have a supported offer type. Cost Management doesn't support management groups with Microsoft Customer Agreement subscriptions.

More information

Supported subscription types are shown at [Understand Cost Management data](#).

Unauthorized

Error message `Unauthorized`.

Mitigation

If using the ExternalBillingAccounts or ExternalSubscriptions APIs, verify that the Microsoft.CostManagement resource providerRP was [registered](#) for your Azure Active Directory instance. Resource Provider registration is required to use Cost Management for AWS.

If you get an `Empty GUID user id` error, update the bearer token associated with the request. You might temporarily see the error in the Azure portal, but it should resolve itself. If you continue to see the error in the Azure portal, refresh your browser.

Also, see [AuthorizationFailed](#).

More information

For more information, see [Set up AWS integration with Cost Management](#).

Create a support request

If you're facing an error not listed above or need more help, file a [support request](#) and specify the issue type as **Billing**.

Next steps

- Read the [Cost Management + Billing frequently asked questions \(FAQ\)](#).

Microsoft Cost Management

Article • 06/07/2022

The Cost Management APIs provide the ability to explore cost and usage data via multidimensional analysis, where creating customized filters and expressions allow you to answer consumption-related questions for your Azure resources. These APIs are currently available for Azure Enterprise customers.

REST Operation Groups

Operation Group	Description
Dimensions	Provides operations to get supported dimensions for your usage under a variety of scopes. You can retrieve a list of dimensions that can be used as inputs for generating queries with the Query or Exports API.
Query	Provides operations to get aggregated cost and usage data based on the query you supply. Data can be filtered, sorted, and grouped by all available Dimensions (accessible through the Dimensions API).
Exports	Provides operations to schedule recurring exports of cost and usage data to blob storage. Data can be filtered, sorted, and grouped by all available Dimensions (accessible through the Dimensions API).

See also

[Azure consumption REST API documentation](#)

Azure Retail Prices overview

Article • 02/16/2023

Azure customers have been looking for a programmatic way to retrieve retail prices for all Azure services. Now you can use the Retail Rates Prices API to get retail prices for all Azure services. Previously, the only way that you could retrieve prices for Azure services was to either use the Azure Pricing Calculator or use the Azure portal. This API gives you an unauthenticated experience to get retail rates for all Azure services. Use the API to explore prices for Azure services against different regions and different SKUs. The programmatic API can also help you create your own tools for internal analysis and price comparison across SKUs and regions.

Important

The currency that Microsoft uses to price all Azure services is USD. Prices shown in USD currency are Microsoft retail prices. Other non-USD prices returned by the API are for your reference to help you estimate budget expenses.

Preview version

You can use the new 2023-01-01-preview API version at

<https://prices.azure.com/api/retail/prices?api-version=2023-01-01-preview>.

The preview version is backward compatible with previous API versions. The URL returns the full set of meters, including primary meters and non-primary meters.

Using either <https://prices.azure.com/api/retail/prices> or

<https://prices.azure.com/api/retail/prices?api-version=2023-01-01-preview> returns the full set of meters.

Rates for savings plans are only available with the

<https://prices.azure.com/api/retail/prices?api-version=2023-01-01-preview> version.

Primary meter filtering

You can limit responses to retrieve only primary meter prices with:

HTTP

```
https://prices.azure.com/api/retail/prices?api-version=2023-01-01-preview&meterRegion='primary'
```

Primary meter filtering is supported by `2021-10-01` and later API versions including `2023-01-01`.

Azure saving plan support

As previously mentioned, Azure savings plan retail prices are only supported by the preview version (<https://prices.azure.com/api/retail/prices?api-version=2023-01-01-preview>). For the meters eligible for a savings plan, results have a corresponding section for savings plan. When you commit to a savings plan, you get a cheaper price than the `OnDemand` price. For more information about Azure savings plans, see [Azure savings plans documentation](#).

Here's an example API response with savings plan information:

JSON

```
{  
    "currencyCode": "USD",  
    "tierMinimumUnits": 0.0,  
    "retailPrice": 2.305,  
    "unitPrice": 2.305,  
    "armRegionName": "southindia",  
    "location": "IN South",  
    "effectiveStartDate": "2019-05-14T00:00:00Z",  
    "meterId": "0084b086-37bf-4bee-b27f-6eb0f9ee4954",  
    "meterName": "M8ms",  
    "productId": "DZH318Z0BQ4W",  
    "skuId": "DZH318Z0BQ4W/00BQ",  
    "availabilityId": null,  
    "productName": "Virtual Machines MS Series",  
    "skuName": "M8ms",  
    "serviceName": "Virtual Machines",  
    "serviceId": "DZH313Z7MMC8",  
    "serviceFamily": "Compute",  
    "unitOfMeasure": "1 Hour",  
    "type": "Consumption",  
    "isPrimaryMeterRegion": true,  
    "armSkuName": "Standard_M8ms",  
    "savingsPlan": [  
        {  
            "unitPrice": 0.8065195,  
            "retailPrice": 0.8065195,  
            "term": "3 Years"  
        },  
        {  
            "unitPrice": 1.5902195,  
            "retailPrice": 1.5902195,  
            "term": "1 Year"  
        }  
    ]  
}
```

```
        }  
    ],  
},
```

API endpoint

<https://prices.azure.com/api/retail/prices>

API sample calls

Here are some examples:

Example calls filtered for only virtual machines

HTTP

```
https://prices.azure.com/api/retail/prices?\$filter=serviceName eq 'Virtual  
Machines'
```

Example calls filtered for only reservations

HTTP

```
https://prices.azure.com/api/retail/prices?\$filter=priceType eq 'Reservation'
```

Example calls filtered for reserved instance virtual machines

HTTP

```
https://prices.azure.com/api/retail/prices?\$filter=serviceName eq 'Virtual  
Machines' and priceType eq 'Reservation'
```

Example calls filtered for compute resources

HTTP

```
https://prices.azure.com/api/retail/prices?\$filter=serviceFamily eq 'Compute'
```

Example calls filtered for compute with currency in euro

HTTP

```
https://prices.azure.com/api/retail/prices?  
currencyCode='EUR'&\$filter=serviceFamily eq 'Compute'
```

API response examples

Here's a sample API response, without reservation prices.

JSON

```
{  
    "currencyCode": "USD",  
    "tierMinimumUnits": 0.0,  
    "retailPrice": 0.176346,  
    "unitPrice": 0.176346,  
    "armRegionName": "westeurope",  
    "location": "EU West",  
    "effectiveStartDate": "2020-08-01T00:00:00Z",  
    "meterId": "000a794b-bdb0-58be-a0cd-0c3a0f222923",  
    "meterName": "F16s Spot",  
    "productId": "DZH318Z0BQPS",  
    "skuId": "DZH318Z0BQPS/00TG",  
    "productName": "Virtual Machines FS Series Windows",  
    "skuName": "F16s Spot",  
    "serviceName": "Virtual Machines",  
    "serviceId": "DZH313Z7MMC8",  
    "serviceFamily": "Compute",  
    "unitOfMeasure": "1 Hour",  
    "type": "DevTestConsumption",  
    "isPrimaryMeterRegion": true,  
    "armSkuName": "Standard_F16s"  
}
```

Here's a sample API response with reservation prices and term in the response.

JSON

```
{  
    "currencyCode": "USD",  
    "tierMinimumUnits": 0.0,  
    "reservationTerm": "1 Year",  
    "retailPrice": 25007.0,  
    "unitPrice": 25007.0,  
    "armRegionName": "southcentralus",  
    "location": "US South Central",  
    "effectiveStartDate": "2020-08-01T00:00:00Z",  
    "meterId": "0016083a-928f-56fd-8eeb-39287dcf676d",  
    "meterName": "E64 v4",  
    "productId": "DZH318Z0D1L7",  
    "skuId": "DZH318Z0D1L7/018J",  
    "productName": "Virtual Machines Ev4 Series",  
    "unitType": "Hour",  
    "unitTypeDescription": "Hour",  
    "unitTypeCode": "Hr",  
    "unitTypeOrder": 1  
}
```

```
        "skuName": "E64 v4",
        "serviceName": "Virtual Machines",
        "serviceId": "DZH313Z7MMC8",
        "serviceFamily": "Compute",
        "unitOfMeasure": "1 Hour",
        "type": "Reservation",
        "isPrimaryMeterRegion": true,
        "armSkuName": "Standard_E64_v4"
    }
```

Here's a sample response with a non-USD currency.

JSON

```
{
    "currencyCode": "EUR",
    "tierMinimumUnits": 0,
    "retailPrice": 0.6176,
    "unitPrice": 0.6176,
    "armRegionName": "westeurope",
    "location": "EU West",
    "effectiveStartDate": "2021-04-01T00:00:00Z",
    "meterId": "000a794b-bdb0-58be-a0cd-0c3a0f222923",
    "meterName": "F16s Spot",
    "productId": "DZH318Z0BQPS",
    "skuId": "DZH318Z0BQPS/00TG",
    "productName": "Virtual Machines FS Series Windows",
    "skuName": "F16s Spot",
    "serviceName": "Virtual Machines",
    "serviceId": "DZH313Z7MMC8",
    "serviceFamily": "Compute",
    "unitOfMeasure": "1 Hour",
    "type": "Consumption",
    "isPrimaryMeterRegion": true,
    "armSkuName": "Standard_F16s"
}
```

API response pagination

The API response provides pagination. For each API request, a maximum of 100 records are returned. At the end of the API response, it has the link to next page. For example:

JSON

```
"NextPageLink": https://prices.azure.com:443/api/retail/prices?
$filter=serviceName%20eq%27Virtual%20Machines%27&$skip=100
```

API property details

Here's all the property details that are a part of the API response.

Field	Example Values	Definition
<code>currencyCode</code>	USD	The currency in which rates are defined and returns prices in USD unless specified.
<code>tierMinimumUnits</code>	0	Minimum units of consumption to avail the price
<code>reservationTerm</code>	1 year	Reservation term – one year or three years
<code>retailPrice</code>	0.176346	Prices without discount
<code>unitPrice</code>	0.176346	
<code>armRegionName</code>	<code>westeurope</code>	Azure Resource Manager region where the service is available. This version only supports prices on Commercial Cloud.
<code>Location</code>	EU West	Azure data center where the resource is deployed
<code>effectiveStartDate</code>	2020-08-01T00:00:00Z	Optional field. Shows the date when the retail prices are effective.
<code>meterId</code>	000a794b-bdb0-58be-a0cd-0c3a0f222923	Unique identifier of the resource
<code>meterName</code>	F16s Spot	Name of the meter
<code>productid</code>	DZH318Z0BQPS	UniquelID of the product
<code>skuId</code>	DZH318Z0BQPS/00TG	UniquelID for the SKU
<code>productName</code>	Virtual Machines FS Series Windows	Product name
<code>skuName</code>	F16s Spot	SKU name
<code>serviceName</code>	Virtual Machines	Name of the service
<code>serviceId</code>	DZH313Z7MMC8	UniquelID of the service
<code>serviceFamily</code>	Compute	Service family of the SKU
<code>unitOfMeasure</code>	1 Hour	How usage is measured for the service

Field	Example Values	Definition
Type	DevTestConsumption	Meter consumption type. Other types are <code>Reservation</code> and <code>Consumption</code> .
<code>isPrimaryMeterRegion</code>	True	Indicates whether the meter region is set as a primary meter or not. Primary meters are used for charges and billing.
<code>armSkuName</code>	Standard_F16s	SKU name registered in Azure
<code>term</code>	3 Years	Term length for an Azure savings plan, associated with <code>savingsPlan</code> information.

API filters

Filters are supported for the following fields:

- `armRegionName`
- `Location`
- `meterId`
- `meterName`
- `productid`
- `skuId`
- `productName`
- `skuName`
- `serviceName`
- `serviceId`
- `serviceFamily`
- `priceType`
- `armSkuName`

You append the filters to the API endpoint, as shown in the API sample calls.

Supported currencies

You append the currency code to the API endpoint, as shown in the API sample call.

Currency code	Detail
USD	US dollar
AUD	Australian dollar

Currency code	Detail
BRL	Brazilian real
CAD	Canadian dollar
CHF	Swiss franc
CNY	Chinese yuan
DKK	Danish krone
EUR	Euro
GBP	British pound
INR	Indian rupee
JPY	Japanese yen
KRW	Korean won
NOK	Norwegian krone
NZD	New Zealand dollar
RUB	Russian ruble
SEK	Swedish krona
TWD	Taiwan dollar

Programmatic Example

The following simple python application employs the API to query the retail spot pricing of a virtual machine of the standard NP20 SKU in the South Central US region. The application filters the output of the query and presents the desired information in a table format.

Python

```
#!/usr/bin/env python3
import requests
import json
from tabulate import tabulate

def build_pricing_table(json_data, table_data):
    for item in json_data['Items']:
        meter = item['meterName']
```

```

        table_data.append([item['armSkuName'], item['retailPrice'],
item['unitOfMeasure'], item['armRegionName'], meter, item['productName']])

def main():
    table_data = []
    table_data.append(['SKU', 'Retail Price', 'Unit of Measure', 'Region',
'Meter', 'Product Name'])

    api_url = "https://prices.azure.com/api/retail/prices?api-version=2021-
10-01-preview"
    query = "armRegionName eq 'southcentralus' and armSkuName eq
'Standard_NP20s' and priceType eq 'Consumption' and contains(meterName,
'Spot')"
    response = requests.get(api_url, params={'$filter': query})
    json_data = json.loads(response.text)

    build_pricing_table(json_data, table_data)
    nextPage = json_data['NextPageLink']

    while(nextPage):
        response = requests.get(nextPage)
        json_data = json.loads(response.text)
        nextPage = json_data['NextPageLink']
        build_pricing_table(json_data, table_data)

    print(tabulate(table_data, headers='firstrow', tablefmt='psql'))

if __name__ == "__main__":
    main()

```

Running this code produces the following output:

Meter	SKU	Retail Price	Unit of Measure	Region
	Meter	Product Name		
	Standard_NP20s	0.828503	1 Hour	southcentralus
	NP20s Spot		Virtual Machines NP Series Windows	
	Standard_NP20s	0.448207	1 Hour	southcentralus
	NP20s Spot		Virtual Machines NP Series	

Next steps

- Learn about other [Cost Management APIs](#).

Price Sheet

Reference

Service: Cost Management

API Version: 2023-03-01

Operations

Download	Gets a URL to download the pricesheet for an invoice. The operation is supported for billing accounts with agreement type Microsoft Partner Agreement or Microso...
Download By Billing Profile	Gets a URL to download the current month's pricesheet for a billing profile. The operation is supported for billing accounts with agreement type Microsoft Partn...

Get started with Cost Management + Billing reporting

Article • 10/19/2022

Cost Management + Billing includes several tools to help you understand, report on, and analyze your invoiced Microsoft Cloud and AWS costs. The following sections describe the major reporting components.

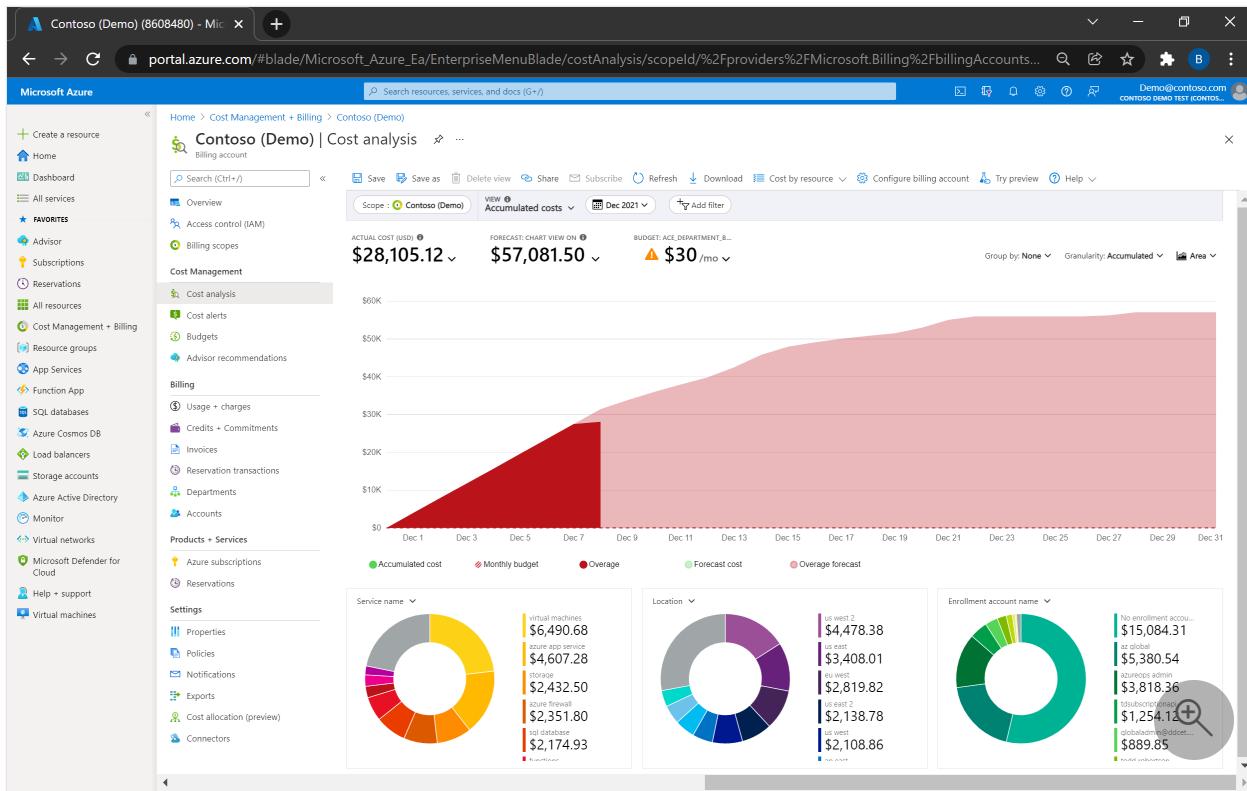
Cost analysis

Cost analysis should be your first stop in the Azure portal when it comes to understanding what you're spending and where you're spending. Cost analysis helps you:

- Visualize and analyze your organizational costs
- Share cost views with others using custom alerts
- View aggregated costs by organization to understand where costs occur over time and identify spending trends
- View accumulated costs over time to estimate monthly, quarterly, or even yearly cost trends against a budget
- Create budgets to provide adherence to financial constraints
- Use budgets to view daily or monthly costs and help isolate spending irregularities

Cost analysis is available from every resource group, subscription, management group, and billing account in the Azure portal. If you manage one of these scopes, you can start there and select **Cost analysis** from the menu. If you manage multiple scopes, you may want to start directly within Cost Management:

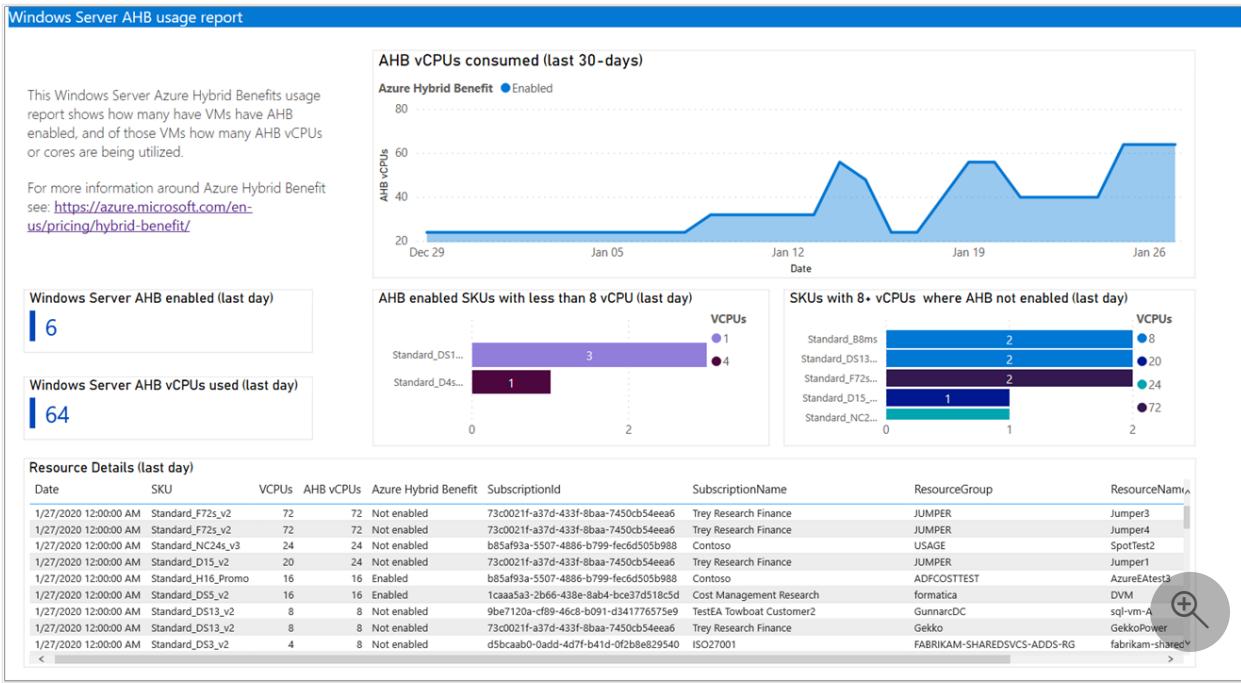
Sign in to the Azure portal > select **Home** in the menu > scroll down under **Tools** and select **Cost Management** > select a scope at the top of the page > in the left menu, select **Cost analysis**.



For more information about cost analysis, see [Explore and analyze costs with cost analysis](#).

Power BI

While cost analysis offers a rich, interactive experience for analyzing and surfacing insights about your costs, there are times when you need to build more extensive dashboards and complex reports or combine costs with internal data. The Cost Management template app for Power BI is a great way to get up and running with Power BI quickly. For more information about the template app, see [Analyze Azure costs with the Power BI App](#).



Need to go beyond the basics with Power BI? The Cost Management connector for Power BI lets you choose the data you need to help you seamlessly integrate costs with your own datasets or easily build out more complete dashboards and reports to meet your organization's needs. For more information about the connector, see [Connect to Cost Management data in Power BI Desktop](#).

Cost details and exports

If you're looking for raw data to automate business processes or integrate with other systems, start by exporting data to a storage account. Scheduled exports allow you to automatically publish your raw cost data to a storage account on a daily, weekly, or monthly basis. With special handling for large datasets, scheduled exports are the most scalable option for building first-class cost data integration. For more information, see [Create and manage exported data](#).

Microsoft Azure

Home > Cost Management + Billing > Contoso (Demo) (8608480)

Search resources, services, and docs (G+)

Contoso (Demo) (8608480) | Exports

Billing account

Overview

Access control (IAM)

Billing scopes

Cost Management

- Cost analysis
- Cost alerts
- Budgets
- Advisor recommendations

Billing

- Usage + charges
- Credits + Commitments
- Invoices
- Reservation transactions
- Departments
- Accounts

Products + Services

- Azure subscriptions
- Reservations + Hybrid Benefit

Settings

- Properties
- Policies
- Notifications
- Exports
- Cost allocation (preview)

How would you rate exporting data on a schedule? →

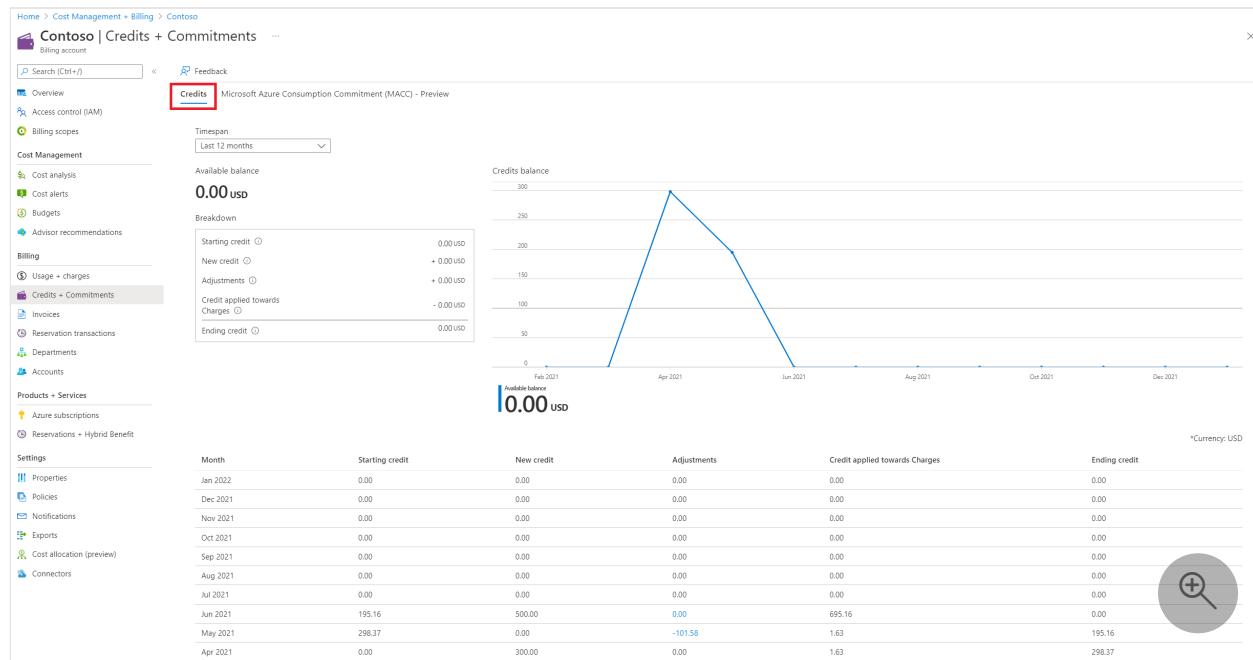
Scope	Status	Last Exported	Frequency	Storage Account	Container	File Type
BAECCMJuly	Active	1/4/2022, 7:11 PM PST	1/5/2022, 10:13 PM PST	Daily	daadatalakegen2	...
EAPartitioningTest	Active	1/4/2022, 7:11 PM PST	1/5/2022, 10:13 PM PST	Daily	cmmexportstests1	...
BA-ECC-CM-History0	Inactive	7/27/2021, 11:56 AM PDT	...	Custom	daadatalakegen2	...
BA-ECC-CM-History1	Inactive	7/27/2021, 11:56 AM PDT	...	Custom	daadatalakegen2	...
BA-ECC-CM-History2	Inactive	7/27/2021, 11:56 AM PDT	...	Custom	daadatalakegen2	...
BA-ECC-CM-History3	Inactive	7/27/2021, 11:56 AM PDT	...	Custom	daadatalakegen2	...
myexporttest	Inactive	9/16/2021, 3:35 PM PDT	...	Custom	currencyblobdag	...
Tip2020-06-01ValidateEaAtEnrollmentScope	Inactive	9/23/2021, 2:47 AM PDT	...	Custom	cmmexportstests1	...
Tip2019-11-01ValidateEaAtEnrollmentScope	Inactive	9/23/2021, 2:57 AM PDT	...	Custom	cmmexportstests1	...
SampleExport1	Active	1/4/2022, 7:10 PM PST	1/5/2022, 10:13 PM PST	Daily	cmmexportstests1	...
Sample2	Active	1/4/2022, 7:09 PM PST	1/5/2022, 10:13 PM PST	Daily	cmmvintegrationtests	...
mnmohammedexport	Inactive	10/6/2021, 8:18 PM PDT	...	Custom	mnmohammedexport	...
SampleExport3	Active	1/4/2022, 7:10 PM PST	1/5/2022, 10:13 PM PST	Daily	cmmvintegrationtests	...
Functional_ValidateEaAtEnrollmentScope_2019-1...	Inactive	10/21/2021, 1:11 PM PDT	...	Custom	cmmexportstests1	...
Functional_ValidateEaAtEnrollmentScope_2020-0...	Inactive	10/21/2021, 1:20 PM PDT	...	Custom	cmmexportstests1	...
MNGoostepot	Inactive	11/2/2021, 1:20 PM PDT	...	Custom	tagtesting1	...
Functional_ValidateEaAtEnrollmentScope_2020-0...	Inactive	11/2/2021, 1:23 PM PDT	...	Custom	cmmexportstests1	...
UMBTEST	Inactive	11/2/2021, 8:18 PM PDT	...	Custom	acmeportugueseblobdag	...
exportcostest21106	Active	1/4/2022, 7:10 PM PST	1/5/2022, 10:13 PM PST	Daily	exportcostest21106	...
EAAdministest2110280040006056	Inactive	11/8/2021, 3:35 AM PST	...	Custom	acmexporttacted	...
Functional_ValidateEaAtEnrollmentScope_2019-1...	Inactive	11/9/2021, 8:13 AM PST	...	Custom	cmmexportstests1	...
Functional_ValidateEaAtEnrollmentScope_2019-1...	Inactive	11/12/2021, 9:24 PM PST	...	Custom	cmmexportstests1	...
Functional_ValidateEaAtEnrollmentScope_2020-0...	Inactive	11/12/2021, 9:37 PM PST	...	Custom	cmmexportstests1	...
Functional_ValidateEaAtEnrollmentScope_2021-0...	Inactive	11/19/2021, 1:30 AM PST	...	Custom	cmmexportstests1	...

If you need more fine-grained control over your data requests, the Cost Details API offers a bit more flexibility to pull raw data the way you need it. For more information, see [Cost Details API](#).

Invoices and credits

Cost analysis is a great tool for reviewing estimated, unbilled charges or for tracking historical cost trends, but it may not show your total billed amount because credits, taxes, and other refunds and charges not available in Cost Management. To estimate your projected bill at the end of the month, start in cost analysis to understand your forecasted costs, then review any available credit or prepaid commitment balance from **Credits or Payment methods** for your billing account or billing profile within the Azure portal. To review your final billed charges after the invoice is available, see **Invoices** for your billing account or billing profile.

Here's an example that shows credits on the Credits tab on the Credits + Commitments page.



For more information about your invoice, see [View and download your Microsoft Azure invoice](#)

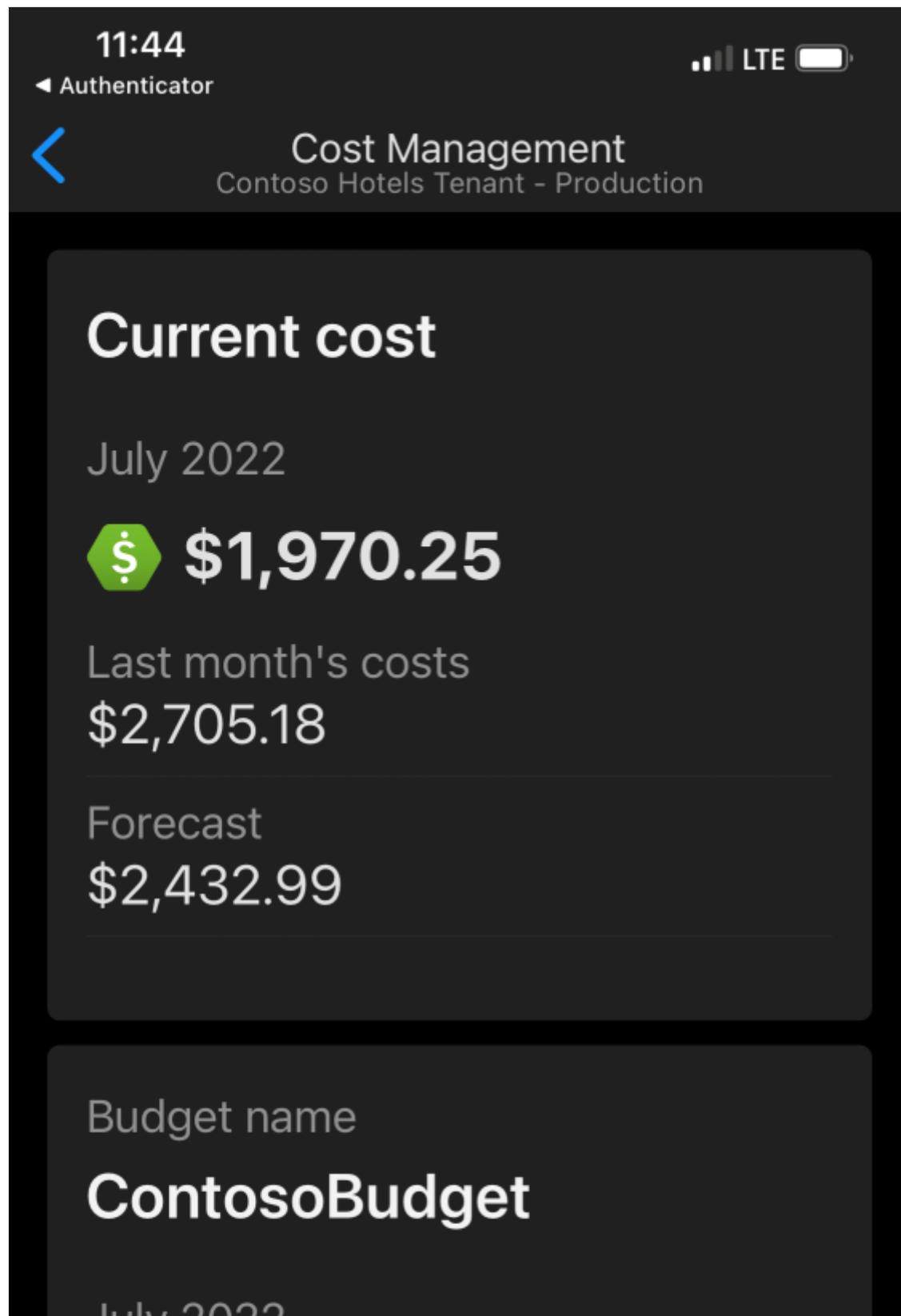
For more information about credits, see [Track Microsoft Customer Agreement Azure credit balance](#).

Microsoft Azure mobile app

With the Azure app, you can keep track of the status of your Azure resources, such as virtual machines (VMs) and web apps, from your mobile device. The app also sends alerts about your environment.

You can also use the Azure app to track the status of subscription or resource group cost. You can see your current cost, last month's cost, forecasted cost, and view your budget usage.

The app is available for [iOS](#) and [Android](#).



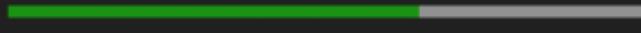
July 2022



\$3,000.00/month

Budget usage

64.4%



Budget remaining

\$1,068.15

Period



Next steps

- Explore and analyze costs with cost analysis.
- Analyze Azure costs with the Power BI App.
- Connect to Microsoft Cost Management data in Power BI Desktop.
- Create and manage exported data.

Quickstart: Start using Cost analysis

Article • 03/12/2023

Before you can control and optimize your costs, you first need to understand where they originated – from the underlying resources used to support your cloud projects to the environments they're deployed in and the owners who manage them. Full visibility backed by a thorough tagging strategy is critical to accurately understand your spending patterns and enforce cost control mechanisms.

In this quickstart, you use Cost analysis to explore and get quick answers about your costs. You can see a summary of your cost over time to identify trends and break costs down to understand how you're being charged for the services you use. For advanced reporting, use Power BI or export raw cost details.

Prerequisites

Cost Management isn't available for classic Cloud Solution Provider and sponsorship subscriptions. For more information about supported subscription types, see [Understand Cost Management data](#).

You must have Read access to use Cost Management. You might need to wait 48 hours to view new subscriptions in Cost Management.

Get started

Cost analysis is your tool for interactive analytics and insights. It should be your first stop when you need to explore or get quick answers about your costs. You explore and analyze costs using *views*. A view is a customizable report that summarizes and allows you to drill into your costs. Cost analysis comes with various built-in views that summarize:

- Cost of your resources at various levels.
- Overarching services spanning all your resources.
- Amortized reservation usage.
- Cost trends over time.

Depending on how you access Cost analysis, you may see two options. If available, we recommend starting with **Cost analysis (preview)** since you can access all views from one central page.

The first time you open Cost analysis, you start with either a list of available cost views or a customizable area chart. This section walks through the list of views. If Cost analysis shows an area chart by default, see [Analyze costs with customizable views](#).

Cost analysis has two types of views: **smart views** that offer intelligent insights and more details by default and **customizable views** you can edit, save, and share to meet your needs. Smart views open in tabs in Cost analysis. To open a second view, select the + symbol to the right of the list of tabs. You can open up to five tabs at one time. Customizable views open outside of the tabs in the custom view editor.

As you explore the different views, notice that Cost analysis remembers which views you've used in the **Recent** section. Switch to the **All views** section to explore all of your saved views and the ones Microsoft provides out of the box. If there's a specific view that you want quick access to, select **Pin to recent** from the **All views** list.

The screenshot shows the Azure Cost Management interface for a management group named 'Trey Research'. The 'All views' tab is selected in the top navigation bar. Below it, there are four recommended views: 'Accumulated costs (classic)', 'Subscriptions', 'Resource groups', and 'Services'. The main list of views is titled 'Recent' and contains several entries, each with a 'Pin to recent' button. One entry, 'Test-view-05', has its 'Pin to recent' button highlighted with a red box. Another entry, 'Reservations', also has its 'Pin to recent' button highlighted with a red box. The list includes views like 'Test-view-01' through 'Test-view-05', 'Product breakdown (classic)', 'Public6MonthView (classic)', 'Testing scheduled email (classic)', 'Test-view-02', 'Test-view-03', 'Test-view-04', 'Accumulated costs (classic)', 'Daily costs (classic)', 'Invoice details (classic)', 'Reservations', 'Resource groups', 'Services', and 'Subscriptions'. Each entry shows its name, group, date range, and filters. At the bottom right of the page is a feedback survey with the question 'How would you rate the cost analysis preview?' and a magnifying glass icon.

Views in the **Recommended** list may vary based on what users most commonly use across Azure.

Analyze costs with smart views

If you're new to Cost analysis, we recommend starting with a smart view, like the Resources view. Smart views include:

- Key performance indicators (KPIs) to summarize your cost
- Intelligent insights about your costs like anomaly detection
- Expandable details with the top contributors
- A breakdown of costs at the next logical level in the resource or product hierarchy

When you first open a smart view, note the date range for the period. Most views show the current calendar month, but some use a different period that better aligns to the goals for the view. As an example, the Reservations view shows the last 30 days by default to give you a clearer picture of reservation utilization over time. To choose a different date range, use the arrows in the date pill to switch to the previous or next period, or select the text to open a menu with other options.

Check the **Total** cost KPI at the top of the page to confirm it matches your expectations. Note the small percentage next to the total – it's the change compared to the previous period. Check the **Average** cost KPI to note whether costs are trending up or down unexpectedly.

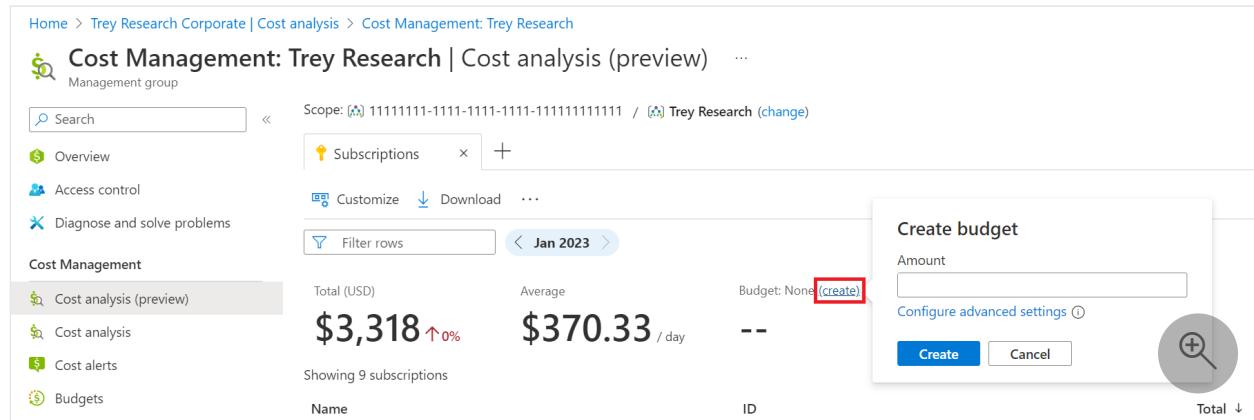
If showing three months or less, the Average cost API compares the cost from the start of the period (up to but not including today) to the same number of days in the previous period. If showing more than three months, the comparison looks at the cost up to but not including the current month.

We recommend checking your cost weekly to ensure each KPI remains within the expected range. If you recently deployed or changed resources, we recommend checking daily for the first week or two to monitor the cost changes.

Note

If you want to monitor your forecasted cost, you can enable the **Forecast KPI preview feature** in Cost Management Labs, available from the [Try preview](#) command.

If you don't have a budget, select the **create** link in the **Budget** KPI and specify the amount you expect to stay under each month. To create a quarterly or yearly budget, select the **Configure advanced settings** link.



Depending on the view and scope you're using, you may also see cost insights below the KPIs. Cost insights show important datapoints about your cost – from discovering

top cost contributors to identifying anomalies based on usage patterns. Select the **See insights** link to review and provide feedback on all insights. Here's an insights example.

The screenshot shows the Azure Cost Management interface for the 'Trey Research Corporate' subscription. On the left, there's a sidebar with various navigation links like Overview, Access control, Diagnose and solve problems, Cost Management, Billing, Products + services, and Support + troubleshooting. The main area displays a summary of costs: Total (USD) \$2,618, Average \$93.51/day, and a budget of None. Below this is a table of 118 resources, with one row highlighted and a red box around the 'See insights' button. To the right, a panel titled 'Insights' lists several items with detailed descriptions and 'Is this helpful?' links. One item is highlighted with a red box: 'mc_analyticsengine_analyticsengine_eastus is 30% of your total cost'. A magnifying glass icon is also present in this panel.

Lastly, use the table to find your top cost contributors and expand each row to understand how costs are broken down to the next level. Examples include resources with their product meters and services with a breakdown of products.

This screenshot shows the 'Cost analysis (preview)' view for the 'Trey Research' management group. The sidebar includes links for Cost analysis, Cost alerts, Budgets, Advisor recommendations, Invoices, Payment methods, Azure subscriptions, Connectors for AWS, and New support request. The main area shows a summary of costs: Total (USD) \$3,318, Average \$370.33/day, and a budget of None. Below this is a table of 9 subscriptions, with two rows expanded: 'Trey Research R&D Playground' and 'Trey Research Corporate'. The 'Trey Research Corporate' row has a red box around the 'See insights' button. To the right, a detailed breakdown of costs is shown in a table with columns for Name, ID, and Total. A large circular callout highlights the entry for 'mc_analyticsengine_analyticsengine_eastus' with a value of '\$246.42'. A magnifying glass icon is also present in this section.

This view is where you spend most of your time in Cost analysis. To explore further:

1. Open other smart views to get different perspectives on your cost.
2. If you want to drill into data further, you might need to [Change scope](#) to a lower level. For example, you can't view the Subscriptions smart view if your current

scope is a subscription.

3. Open a custom view and apply other filters or group the data to explore.

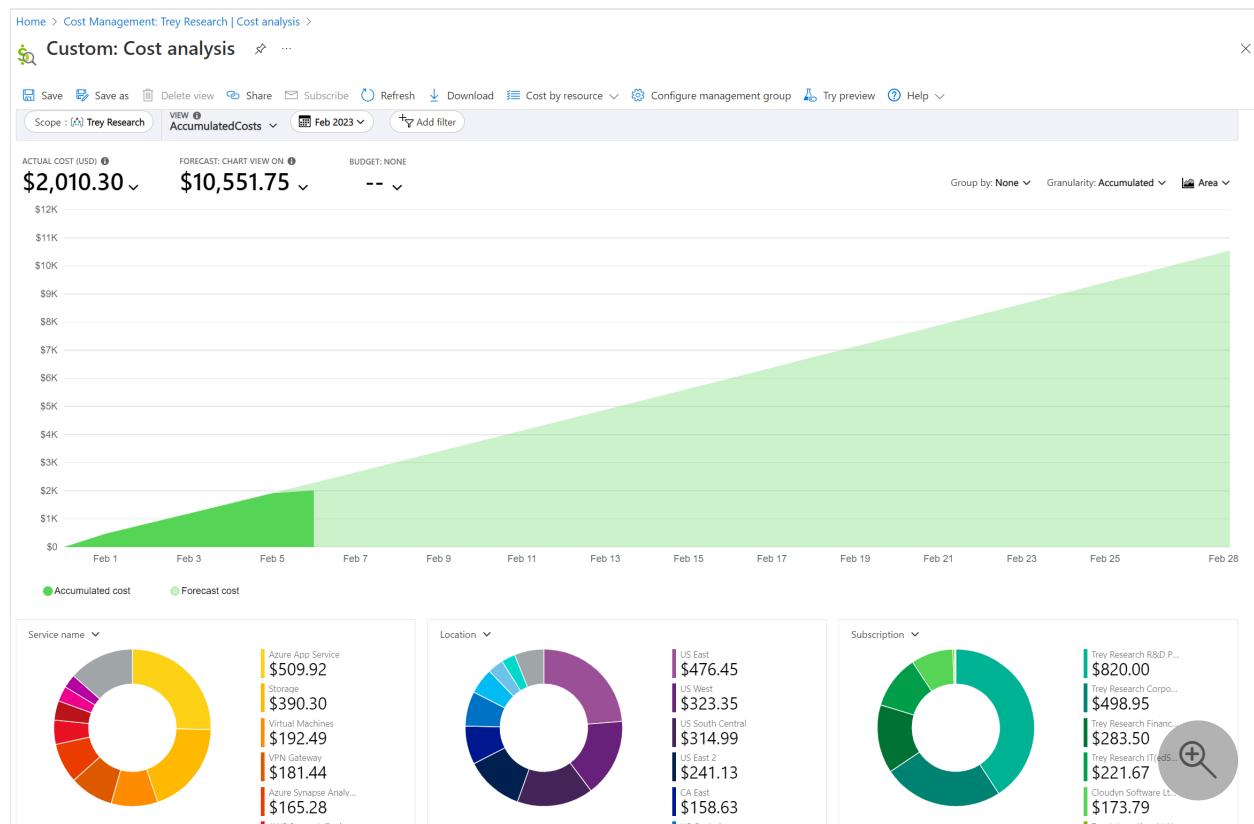
① Note

If you want to visualize and monitor daily trends within the period, enable the **chart preview feature** in Cost Management Labs, available from the **Try preview** command.

Analyze costs with customizable views

While smart views offer a highly curated experience for targeted scenarios, custom views allow you to drill in further and answer more specific questions. Like smart views, custom views include a specific date range, granularity, group by, and one or more filters. Five custom views are provided for you to show how costs change over time. They're separated by resource and product. All aspects of custom views can be changed to help answer simple questions. If you require more advanced reporting, like grouping by multiple attributes or fully customizable reports, use Power BI or export raw cost details.

Here's an example of the Accumulated Costs customizable view.



After you customize your view to meet your needs, you may want to save and share it with others. To share views with others:

1. Save the view on a subscription, resource group, management group, or billing account.
2. Share a URL with view configuration details, which they can use on any scope they have access to.
3. Ping the view to an Azure portal dashboard. Pinning requires access to the same scope.
4. Download an image of the chart or summarized cost details in an Excel or CSV file.
5. Subscribe to scheduled alerts on a daily, weekly, or monthly basis.

All saved views are available from the **All views** list discussed previously.

Download cost details

While all smart and custom views can be downloaded, there are a few differences between them.

Customizable chart views are downloaded as an image, smart views aren't. To download an image of the chart, use customizable views.

When you download table data, smart views include an extra option to include nested details. There are a few extra columns available in smart views. We recommend starting with smart views when you download data.

The screenshot shows the Azure Cost Management + Billing interface. The main area displays cost summary information: Total (CAD) CA\$14.60, Average CA\$0.47 / day, and Budget: None (0%). Below this, it shows 'Showing 2 subscriptions' with two entries: 'testsub-1011' and 'testsub-1011-02'. To the right, a 'Download' pane is open, titled 'Cost analysis'. It includes sections for 'Download a file', 'Open in Power BI', and 'Automate the download'. Under 'Select data you want to download *', there are two radio button options: 'Subscriptions only' (which is selected) and 'Subscriptions with resource groups'. At the bottom of the pane is a large blue 'Download' button.

Although Power BI is available for all Microsoft Customer Agreement billing profiles and Enterprise Agreement billing accounts, you only see the option from the smart view Download pane when using a supported scope.

Cost Management: Contoso billing account | Cost analysis (preview)

Scope: Contoso billing account (change)

Subscriptions

Total (CAD) CA\$14.60 ↑0% Average CA\$0.47 / day

Showing 2 subscriptions

Name	ID
> testsub-1011	11
> testsub-1011-02	11

Download

Cost analysis

Download a file Open in Power BI Automate the download

Type of data: Template app

Instructions:

1. Install the Cost Management app
2. Open the app and click the Connect your data link at the top
3. Copy the scope ID below and paste it into the BillingProfileIdOrEnrollmentNumber field
4. Set the Scope field to Enrollment Number
5. Finish installing the app

Scope ID: 8980343

Copy Learn more

Regardless of whether you start on smart or customizable views, if you need more details, we recommend that you export raw details for full flexibility. Smart views include the option under the **Automate the download** section.

Cost Management: Contoso billing account | Cost analysis (preview)

Scope: Contoso billing account (change)

Subscriptions

Total (CAD) CA\$14.60 ↑0% Average CA\$0.47 / day

Showing 2 subscriptions

Name	ID
> testsub-1011	11
> testsub-1011-02	11

Download

Cost analysis

Download a file Open in Power BI **Automate the download**

Recurring export to storage

Configure exports to automatically publish your billing data to a storage account on daily, weekly, or monthly basis.

Schedule an export

Integrate with the API

GET https://management.azure.com/providers/Microsoft.Billing/billingAccounts/8980343/providers/Microsoft.Consumption/usagedetails?api-version=2019-10-01

Copy Learn how to use the Usage Details API

Understand your forecast

Forecast costs are available from both smart and custom views. In either case, the forecast is calculated the same way based on your historical usage patterns for up to a year in the future.

Your forecast is a projection of your estimated costs for the selected period. Your forecast changes depending on what data is available for the period, how long of a period you select, and what filters you apply. If you notice an unexpected spike or drop in your forecast, expand the date range and use grouping to identify large increases or decreases in historical cost. You can filter them out to normalize the forecast.

When you select a budget in a custom view, you can also see if or when your forecast would exceed your budget.

More information

For more information about using features in costs analysis, see the following articles:

- For built-in views, see [Use built-in views in Cost analysis](#).
- To learn more about customizing views, see [Customize views in cost analysis](#).
- Afterward you can [Save and share customized views](#).

If you need advanced reporting outside of cost analysis, like grouping by multiple attributes or fully customizable reports, you can use:

- [Power BI Desktop](#)
- [Cost Management Power BI App](#)
- Usage data from exports or APIs
 - See [Choose a cost details solution](#) to help you determine if exports from the Azure portal or if cost details from APIs are right for you.

Be sure to [configure subscription anomaly alerts](#) and set up a [budget](#) to help drive accountability and cost control.

Next steps

Advance to the first tutorial to learn how to create and manage budgets.

[Create and manage budgets](#)

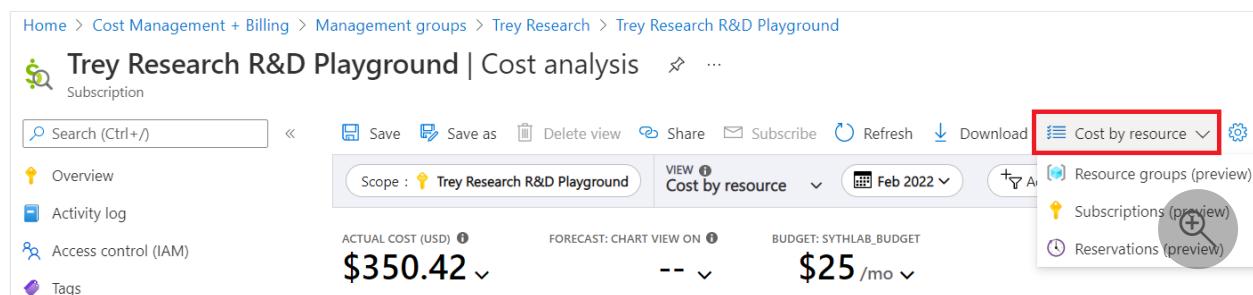
Use built-in views in Cost analysis

Article • 09/09/2022

Cost Management includes several tools to help you view and monitor your cloud costs. As you get started, cost analysis is the first one you should familiarize yourself with. And within cost analysis, you'll start with built-in views. This article helps you understand when to use which view, how each one provides unique insights about your costs and recommended next steps to investigate further.

Access built-in views

When you're in classic Cost analysis, you can access the preview views at the top of the page with the **Cost by resource** list.



The screenshot shows the Microsoft Cost Management portal interface. At the top, there's a breadcrumb navigation: Home > Cost Management + Billing > Management groups > Trey Research > Trey Research R&D Playground. Below the navigation is the title "Trey Research R&D Playground | Cost analysis". On the left, there's a sidebar with links: Overview, Activity log, Access control (IAM), and Tags. The main area has three summary metrics: ACTUAL COST (USD) \$350.42, FORECAST: CHART VIEW ON --, and BUDGET: SYTHLAB_BUDGET \$25 /mo. Above these metrics is a "Scope : Trey Research R&D Playground" dropdown. To the right of the metrics is a "VIEW" dropdown set to "Cost by resource". Further right are buttons for "Feb 2022" and "Download". A "Cost by resource" button is highlighted with a red box. A dropdown menu is open from this button, listing "Resource groups (preview)", "Subscriptions (preview)", and "Reservations (preview)".

Analyze resource costs

Cost Management offers two views to analyze your resource costs:

- **Cost by resource**
- **Resources (preview)**

Both views are only available when you have a subscription or resource group scope selected.

The classic **Cost by resource** view shows a list of all resources. Information is shown in tabular format.

Home > Cost Management: Trey Research R&D Playground > Cost Management: Trey Research R&D Playground

Cost Management: Trey Research R&D Playground | Cost analysis

Subscription

Search (Ctrl+ /) Save Save as Delete view Share Subscribe Refresh Download Cost by resource Configure subscription Try preview Help

Scope : **Trey Research R&D Playground** VIEW Cost by resource Feb 2022 Add filter

ACTUAL COST (USD) FORECAST: CHART VIEW ON BUDGET: SVTHLAB_BUDGET

\$395.28 -- **\$25/mo**

Group by: Resource Granularity: None Table

Filter items 96 rows

Resource	Resource type	Location	Resource group name	Tags	Cost
✓ screenerenv	App Service Environment	us south central	webscreener	env:prod org:trey	\$114.32
	Service name	Service tier	Meter	Cost ↑	
	azure app service	azure app service isolated plan	stamp fee		\$114.32
> screenfarm	App Service plan	us south central	webscreener	projectsilverscreen env:dev or...	\$48.00
✓ kayotest	Azure Database for MySQL server	us east	clancy/test	env:prod org:trey clancytag:cla...	\$30.50
	Service name	Service tier	Meter	Cost ↑	
	azure database for mysql	azure db for mysql sing gen pur-compute g5	vcore		\$27.68
	advanced threat protection	microsoft def for mysql	standard nodes		\$1.60
	azure database for mysql	azure db for mysql sing gen pur-storage	data stored		\$1.22
> screenerenv	App Service Environment	us south central	webscreener	env:prod org:trey	\$24.00
> data1	Disk	fr central	devops	env:prod org:trey	\$18.76
> zmitestplayground	Machine learning	us east 2	devtestlab	--	\$17.88
> clancytest-vnet-2-gw	Virtual network gateway	us central	clancy/test	env:prod clancytag:clancytestrg...	\$15.20
> azureworkshopvmtest_datadisk_0	Disk	us west 2	test_azurelighthouse	env:prod org:trey	\$14.62
> azureworkshopvmtest_datadisk_1	Disk	us west 2	test_azurelighthouse	env:prod org:trey	\$14.62
> stor simple1200	StorSimple Device Manager	us west	clancy/test	env:prod org:trey clancytag:cla...	\$13.69

The preview **Resources** view shows a list of all resources, including deleted resources. The view is like the Cost by resource view in classic cost analysis. Compared to the classic Cost by resource view, the new view:

- Has optimized performance and loads resources faster. It better groups together related costs. Azure and Marketplace costs are grouped together.
- Provides improved troubleshooting details.
- Shows grouped Azure and Marketplace costs together per resource.
- Shows resource types with icons.
- Includes a simpler custom date range selection with support for relative date ranges.
- Allows you to customize the download to exclude nested details. For example, resources without meters in the Resources view.
- Provides smart insights to help you better understand your data, like subscription cost anomalies.

Use either view to:

- Identify top cost contributors by resource.
- Understand how you're charged for a resource.
- Find the biggest opportunities to save money.
- Stop or delete resources that shouldn't be running.
- Identify significant month-over-month changes.
- Identify and tag untagged resources.

Cost Management: Trey Research R&D Playground | Cost analysis (preview)

Subscription

Search (Ctrl+ /) Scope: **Trey Research R&D Playground (change)**

Overview Access control Diagnose and solve problems

Cost Management

- Cost analysis (preview)
- Cost analysis
- Cost alerts
- Budgets
- Advisor recommendations
- Billing
- Usage + charges
- Credits
- Products + services
- Azure subscriptions
- Reservations + Hybrid Benefit
- Settings
- Configuration
- Exports
- Cost allocation (preview)
- Connectors for AWS
- Support + troubleshooting
- New support request

Showing 91 resources No anomalies detected See insights

Name	Type	Resource group	Location	Subscription	Tags	Total
screenerenv	App Service environment	webscreener	us south central	trey research r&d playground	env: prod org: trey	\$114.32
Service	Tier	Meter	Part #			Total ↓
azure app service	azure app service isolated plan	stamp fee	aaa-43476			\$114.32
screenerfarm	App Service plan	webscreener	us south central	trey research r&d playground	project: silverscreen env: trey	\$48.00
Service	Tier	Meter	Part #			Total ↓
azure database for mysql	azure db for mysql sing gen pur-compu	vcore	aaa-90574			\$27.68
advanced threat protection	microsoft def for mysql	standard nodes	aah-05456			\$1.60
azure database for mysql	azure db for mysql sing gen pur-storage	data stored	aaa-90229			\$1.22
clancytest	MySQL server	clancytest	us east	trey research r&d playground	env: prod clancytag: clan	\$30.50
Service	Tier	Meter	Part #			Total ↓
screenerenv	App Service environment	webscreener	us south central	trey research r&d playground	env: prod org: trey	\$24.00
data1	Disk	devops	fr central	trey research r&d playground	env: prod org: trey	\$18.76
zmitestplayground	Machine Learning	devtestlab	us east 2	trey research r&d playground	--	\$17.88
clancytest-net-2-g	Virtual network gateway	clancytest	us central	trey research r&d playground	env: prod org: trey clan	\$15.20
Service	Tier	Meter	Part #			Total ↓
vpn gateway	vpn gateway	vpngw1	aaa-21819			\$15.20
azureworkshopvmt	Disk	test_azurelighthouse	us west 2	trey research r&d playground	env: prod org: trey	\$14.62
azureworkshopvmt	Disk	test_azurelighthouse	us west 2	trey research r&d playground	env: prod org: trey	\$14.62

Analyze resource group costs

The **Resource groups** view separates each resource group in your subscription, management group, or billing account showing nested resources.

Use this view to:

- Identify top cost contributors by resource group.
- Find the biggest opportunities to save money.
- Help perform chargeback by resource group.
- Identify significant month-over-month changes.
- Identify and tag untagged resources using resource group tags.

Cost Management: Trey Research R&D Playground | Cost analysis (preview)

Scope: **Trey Research R&D Playground** (change)

Subscription

Showing 21 resource groups | No anomalies detected | See insights

Name **Subscription** **Total ↓**

Name	Subscription	Total
webscreener	trey research r&d playground	\$186.32
clancytest	trey research r&d playground	\$70.47

Name **Type** **Location** **Tags** **Total ↓**

Name	Type	Location	Tags	Total
screenerenv	App Service environment	us south central	env: prod org: trey	\$114.32
screenerfarm	App Service plan	us south central	project: silverscreen env: dev org: trey	\$48.00
screenerenv	App Service environment	us south central	env: prod org: trey	\$24.00
webscreener / dev	App Service deployment slot	us south central	env: prod org: trey	<\$0.01
webscreener	App Service web app	us south central	project: silverscreen env: dev org: trey	<\$0.01

Name	Type	Location	Tags	Total
kayotest	MySQL server	us east	env: prod org: trey	\$30.50
clancytest-vnet-2-gw	Virtual network gateway	us central	env: prod org: trey clancytag: clancytestrg	\$15.20
storisimple1200	StorSimple device manager	us west	env: prod org: trey clancytag: clancytestrg	\$13.69
storisimplevol	Disk	us west	env: prod org: trey clancytag: clancytestrg	\$2.54
clancystore	Storage account	us central, intercontinental	env: prod org: trey clancytag: clancytestrg	\$1.89
aftestvm_osdisk_1_5c65538192f84b66ad2c	Disk	us central	env: prod org: trey clancytag: clancytestrg	\$1.14
clancyvm1_osdisk_1_bbab08329d04ee2ac	Disk	us central	env: prod org: trey clancytag: clancyv1	\$1.14
azurefiletesttccprivateendpoint	Private endpoint	us central	env: prod projectowner: terryclancy org: tr	\$0.80
clancytestsaendpoint	Private endpoint	us west 2	env: prod org: trey clancytag: clancytestrg	\$0.80
clancytestssprivateendpoint	Private endpoint	us west 2	env: prod org: trey clancytag: clancytestrg	\$0.80
1200testvm_osdisk_1_87430710ea0546838	Disk	us west	env: storisimple org: trey clancytag: clancy	\$0.68

Analyze your subscription costs

The **Subscriptions** view is only available when you have a billing account or management group scope selected. The view separates costs by subscription and resource group.

Use this view to:

- Identify top cost contributors by subscription.
- Find the biggest opportunities to save money.
- Help perform chargeback by resource group.
- Identify significant month-over-month changes.
- Identify and tag untagged resources using resource subscription tags.

The screenshot shows the Azure Cost Management interface for the 'Trey Research' management group. The main view displays a total cost of \$17,730 for 11 subscriptions. Two specific sections are expanded: 'trey research finance' and 'trey research r&d playground'. Each section shows a breakdown of costs by resource name. A search bar at the top left and a magnifying glass icon on the right side of the interface are also visible.

Name	ID	Total
trey research finance	11111111-1111-1111-1111-111111111111	\$9,756
leap		\$9,731
gekko		\$24.14
mc_templar_templar_northeurope		\$0.53
cloud-shell-storage-westus		\$0.35
celticqueen		\$0.22
databricks-rg-fiscalinsights-wuxozk63kt2y		<\$0.01
treycollab		<\$0.01
azureapp-auto-alerts-0c5f05-micflan_microsoft_com		\$0.00

Name	ID	Total
webscreener	11111111-1111-1111-1111-111111111111	\$1,831
clancytest		\$698.11
test_azurelighthouse		\$316.48
adamhourlyexporttest		\$242.84
devops		\$212.78
devtestlab		\$178.57

Review reservation resource utilization

The **Reservations** view provides a breakdown of amortized reservation costs, allowing you to see which resources are consuming each reservation.

The view shows amortized cost for the last 30 days with a breakdown of the resources that utilized each reservation during that time. Any unused portion of the reservation is also available when viewing cost for billing accounts and billing profiles.

Use this view to:

- Identify under-utilized reservations.
- Identify significant month-over-month changes.
- Help perform chargeback for reservations.

Understand amortized costs

Amortized cost breaks down reservation purchases into daily chunks and spreads them over the duration of the reservation term. For example, instead of seeing a \$365 purchase on January 1, you'll see a \$1.00 purchase every day from January 1 to December 31. In addition to basic amortization, these costs are also reallocated and associated by using the specific resources that used the reservation. For example, if that \$1.00 daily charge was split between two virtual machines, you'd see two \$0.50 charges for the day. If part of the reservation isn't utilized for the day, you'd see one \$0.50

charge associated with the applicable virtual machine and another \$0.50 charge with a charge type of UnusedReservation. Unused reservation costs can be seen only when viewing amortized cost.

Because of the change in how costs are represented, it's important to note that actual cost and amortized cost views will show different total numbers. In general, the total cost of months with a reservation purchase will decrease when viewing amortized costs, and months following a reservation purchase will increase. Amortization is available only for reservation purchases and doesn't apply to any other purchases.

Name	ID	Used	Unused	Total
reservation01	11111111-1111-1111-1111-111111111111	\$2.94	\$55.31	\$58.24
sqldb_reservation	11111111-1111-1111-1111-111111111111	\$50.30	\$0.62	\$50.91
vm_reservation	11111111-1111-1111-1111-111111111111	\$42.55	--	\$42.55
Unused reservation	11111111-1111-1111-1111-111111111111	--	\$27.13	\$27.13
Unused reservation	11111111-1111-1111-1111-111111111111	--	\$23.67	\$23.67
Unused reservation	11111111-1111-1111-1111-111111111111	--	\$19.43	\$19.43

Break down product and service costs

The **Services view** shows a list of your services and products. This view is like the Invoice details view in classic cost analysis. The main difference is that rows are grouped by service, making it simpler to see your total cost at a service level. It also separates individual products you're using in each service.

Use this view to:

- Identify top cost contributors by service.
- Find the biggest opportunities to save money.

Home > Cost Management + Billing > Cost Management: Trey Research

Cost Management: Trey Research | Cost analysis (preview)

Management group

Search (Ctrl+ /) Scope: 11111111-1111-1111-1111-111111111111 / Trey Research (change)

Overview Access control Diagnose and solve problems

Customize Download ... Filter rows Feb 2022

Cost Management

- Cost analysis (preview)
- Cost analysis
- Cost alerts
- Budgets
- Advisor recommendations

Billing

- Invoices
- Payment methods

Products + services

- Azure subscriptions
- Reservations + Hybrid Benefit

Settings

- Configuration
- Exports
- Cost allocation (preview)
- Connectors for AWS

Support + troubleshooting

- New support request

Total (USD) **\$6,382** Showing 43 services

Service	Publisher type	Total
azure app service	azure	\$3,200
vpn gateway	azure	\$1,135
aws directory service	aws	\$379.56

Tier Meter Part # Charge type Total

azure app service isolated plan	stamp fee	aaa-43481, aaa-43476	usage	\$2,341
azure app service isolated plan	i2	aaa-43475	usage	\$471.60
azure app service isolated plan	front end	aaa-31827	usage	\$235.80
azure app service std plan	s1	t2x-00094	usage	\$141.48
azure app service shared plan	shared	aaa-41571	usage	\$10.20
azure app service free plan, azu	f1	t2x-00008, Aad-20314	usage	\$0.00

Tier Meter Part # Charge type Total

vpn gateway	vpngw3	aaa-21820	usage	\$985.00
vpn gateway	vpngw1	aaa-21819	usage	\$149.78

aws \$379.56

Review current cost trends

Use the **Accumulated costs** view to:

- Determine whether your current month's costs are on track with your expectations.
- For example, forecast, budget, and credit.

Home > Cost Management: Trey Research R&D Playground

Cost Management: Trey Research R&D Playground | Cost analysis

Subscription

Search (Ctrl+ /) Save Share Delete view Configure subscription Try preview Help

Scope: Trey Research R&D Playground VIEW: Accumulated costs Feb 2022 Add filter

ACTUAL COST (USD) **\$1,515.31** FORECAST: CHART VIEW ON **\$3,398.31** BUDGET: TEST_BUDGET **▲ \$1,300 /mo**

Group by: None Granularity: Accumulated Area

Accumulated cost Monthly budget Overage Forecast cost Overage forecast

Service	Cost
azure app service	\$767.57
storage	\$380.92
azure database for ...	\$136.73
vpn gateway	\$58.14
storisnap	\$52.39

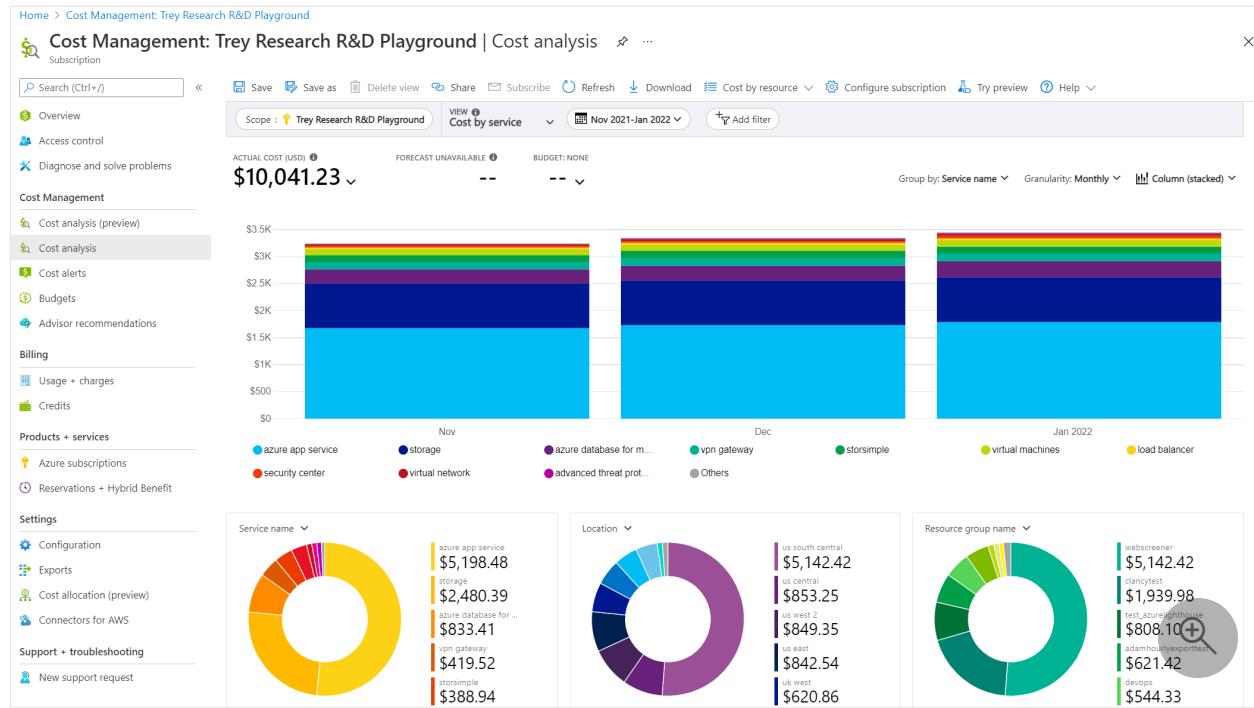
Location	Cost
us south central	\$712.67
us central	\$198.46
us west 2	\$128.27
us east	\$117.74
uk west	\$94.18

Resource Group	Cost
webscreener	\$712.67
cloudbreak	\$271.01
test_azurelighthouse	\$122.56
admincloudview	\$94.27
devops	\$82.59

Compare monthly service run rate costs

Use the **Cost by service** view to:

- Review month-over-month changes in cost.



Reconcile invoiced usage charges

Use the **Invoice details** view to:

- Review and reconcile billed charges.

The screenshot shows the Azure Cost Management interface for the 'Trey Research' management group. The main view displays a table of invoice details for January 2022. The table has columns for Publisher type, Charge type, Service name, Service tier, Meter, and Cost. The table shows 174 rows of data. A summary at the bottom right indicates a total cost of \$15,648.69. The table includes entries for various Azure services like azure app service, vpn gateway, storage, and virtual machines, along with AWS services like aws directory service and premium SSD managed disks. A circular callout highlights a specific row for a load balancer with a cost of \$185.45.

Next steps

- Now that you're familiar with using built-in views, read about [Saving and sharing customized views](#).
- Learn about how to [Customize views in cost analysis](#)

Customize views in cost analysis

Article • 09/09/2022

This article helps you customize views in cost analysis to understand how you're being charged and to investigate unexpected changes.

Prerequisites

To customize views, you must have at least the Cost Management Reader (or Contributor) role.

You should be familiar with the information at [Quickstart: Explore and analyze costs with cost analysis](#).

Get started with customizing views

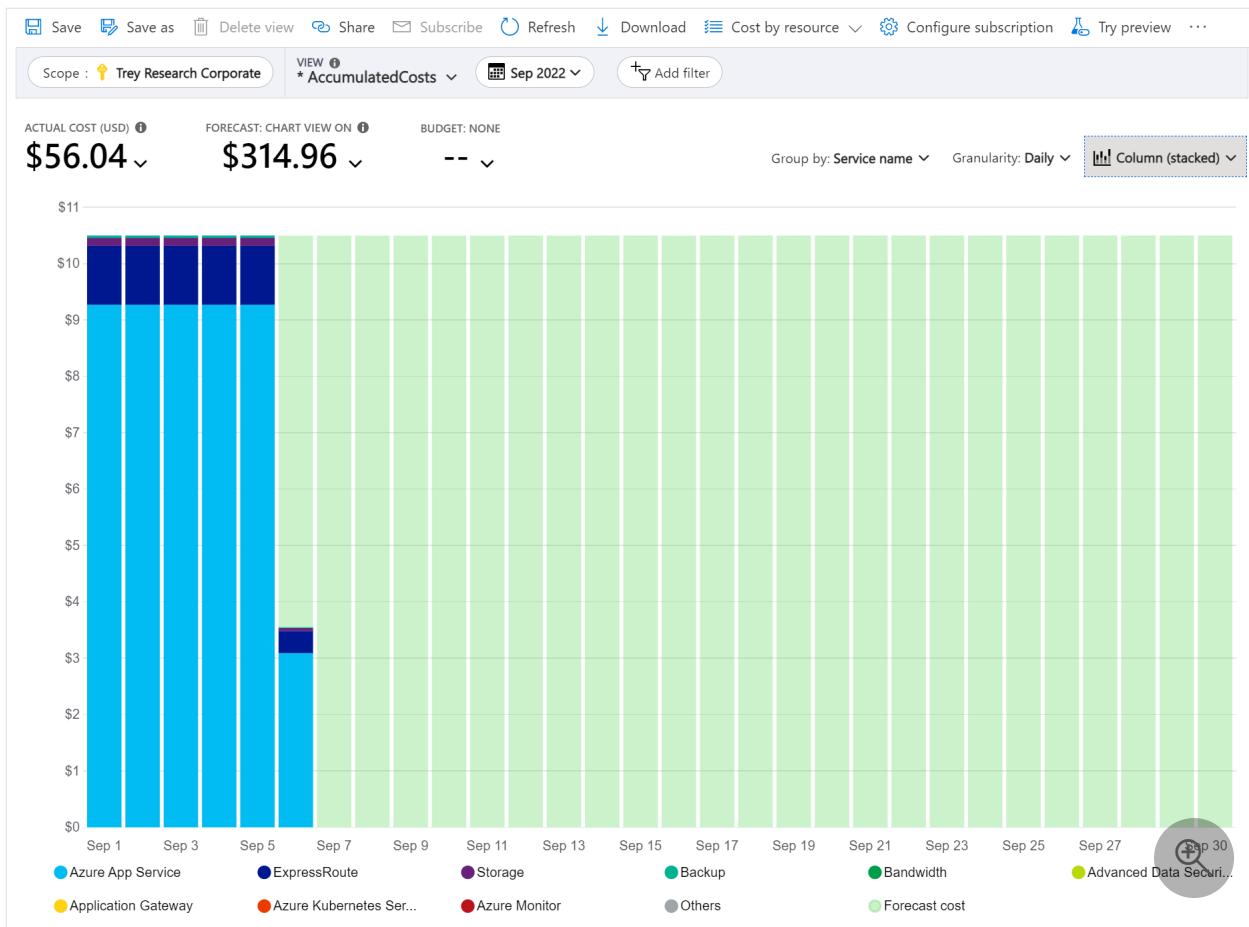
Customizing views in cost analysis includes anything from tweaking display settings to changing what data gets included or how it's summarized. You customize views when trying to understand what you're spending and where the costs originated. For example, you can drill into data, apply specific filters or groupings, or change display settings, like whether to view a chart or table. The following sections cover each of these customization options.

Group costs

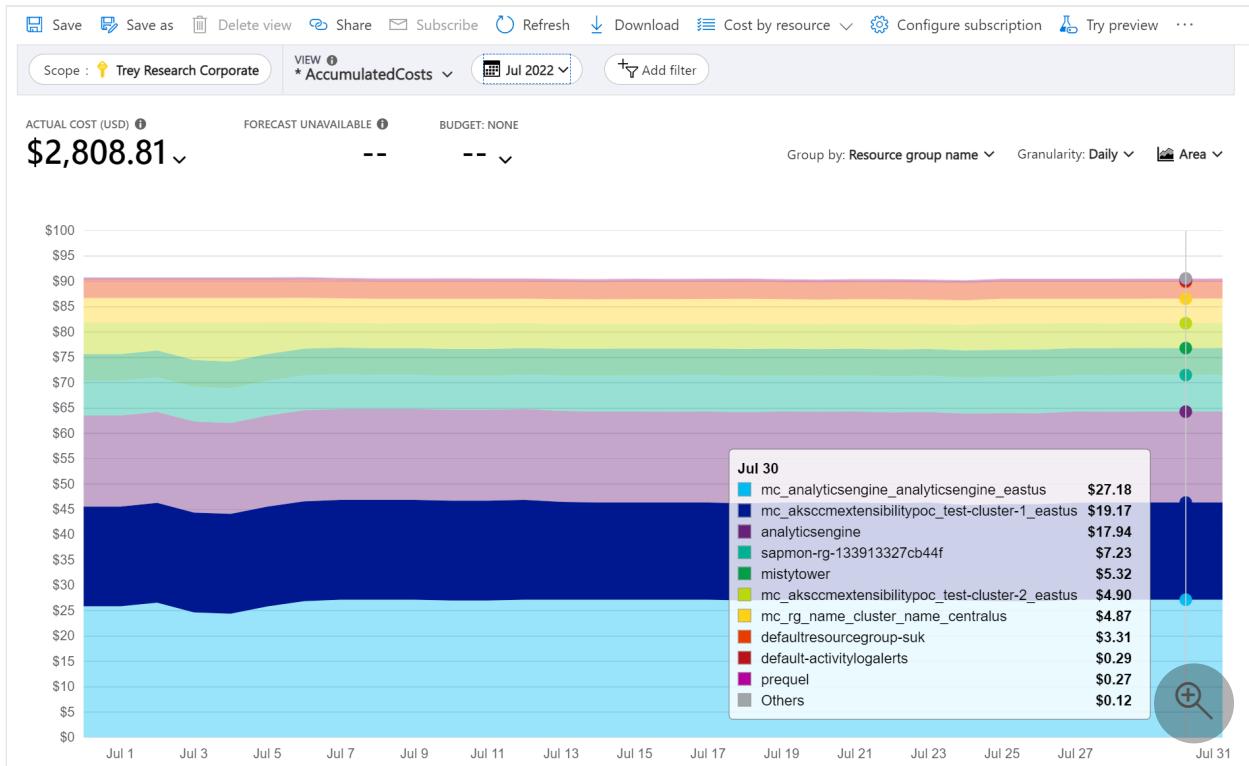
Use the **Group by** option to group common properties so that you get a break down of costs and to identify top contributors. It should be your first change when drilling into data because it helps you identify the largest changes. To group by resource tags, for example, select the tag key you want to group by. Costs are broken down by each tag value, with an extra segment for resources that don't have that tag applied.

Most Azure resources support tagging. However, some tags aren't available in Cost Management and billing. Additionally, resource group tags aren't supported. Support for tags applies to usage reported *after* the tag was applied to the resource. Tags aren't applied retroactively for cost rollups.

Here's a view of Azure service costs for the current month, grouped by Service name.



The following image shows resource group names. You can group by tag to view total costs per tag or group by **Resource group name**.



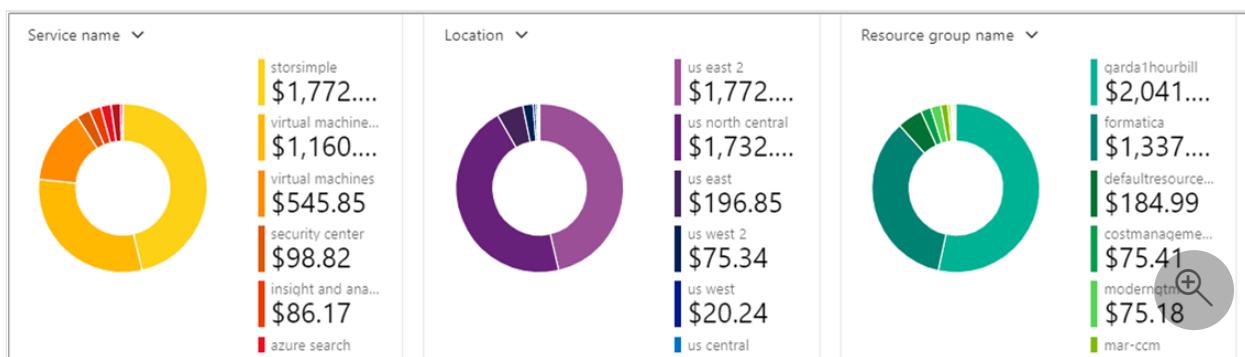
When you're grouping costs by a specific attribute, the top 10 cost contributors are shown from highest to lowest. If there are more than 10, the top nine cost contributors are shown with an **Others** group that represents all remaining groups combined. When

you're grouping by tags, an **Untagged** group appears for costs that don't have the tag key applied. **Untagged** is always last, even if untagged costs are higher than tagged costs. Untagged costs will be part of **Others**, if 10 or more tag values exist. To view what's grouped into **Others**, either select that segment to apply a filter or switch to the table view and change granularity to **None** to see all values ranked from highest to lowest cost.

Classic virtual machines, networking, and storage resources don't share detailed billing data. They're merged as **Classic services** when grouping costs.

Cost analysis doesn't support grouping by multiple attributes. To work around it, you can apply a filter for a desired attribute and group by the more detailed attribute. For instance, filter down to a specific resource group, then group by resource.

Pivot charts under the main chart show different groupings, which give you a broader picture of your overall costs for the selected time period and filters. Select a property or tag to view aggregated costs by any dimension.



Select a date range

There are many cases where you need deeper analysis. Customization starts at the top of the page, with the date selection.

Cost analysis shows data for the current month by default. Use the date selector to switch to common date ranges quickly. Examples include the last seven days, the last month, the current year, or a custom date range. Pay-as-you-go subscriptions also include date ranges based on your billing period, which isn't bound to the calendar month, like the current billing period or last invoice.

[Subscribe](#)[Refresh](#)[Download](#)[Cost by resource](#)

edCosts ▾

Jul 2022 ▾

Add filter

< PREVIOUS

NEXT >

BUD

Recommended

Last 7 days

Aug 31-Sep 6

This month

Sep 2022

Custom date range >

Relative dates

Last 7 days

Aug 31-Sep 6

Last 30 days

Aug 8-Sep 6

Calendar months

This month

Sep 2022

This quarter

Jul-Sep 2022

This year

2022

Last month

Aug 2022

Last quarter

Apr-Jun 2022

Last 3 months

Jun-Aug 2022

Last 6 months

Mar-Aug 2022

Last 12 months

Sep 2021-Aug 2022

Custom date range >



Filter charges

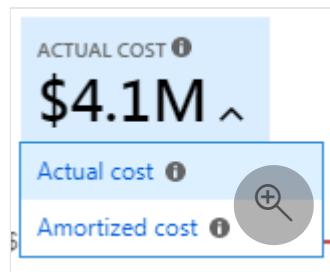
Add filters to narrow down or drill into your specific charges. It's especially helpful when trying to understand an unexpected change. Start by selecting the **Add filter** pill, then select the desired attribute, and lastly select the options you want to filter down to. Your view will automatically update once you've applied the filter.

You can add multiple filters. As you add filters, you'll notice that the available values for each filter include the previously selected filters. For instance, if you apply a resource group filter, then add a resource filter, the resource filter options will only show resources in the selected resource group.

When you view charts, you can also select a chart segment to apply a filter. After selecting a chart segment, you should consider changing the group by attribute to see other details about the attribute you selected.

Switch between actual and amortized cost

By default, cost analysis shows all usage and purchase costs as they're accrued and will show on your invoice, also known as **Actual cost**. Viewing actual cost is ideal for reconciling your invoice. However, purchase spikes in cost can be alarming when you're keeping an eye out for spending anomalies and other changes in cost. To flatten out spikes caused by reservation purchase costs, switch to **Amortized cost**.



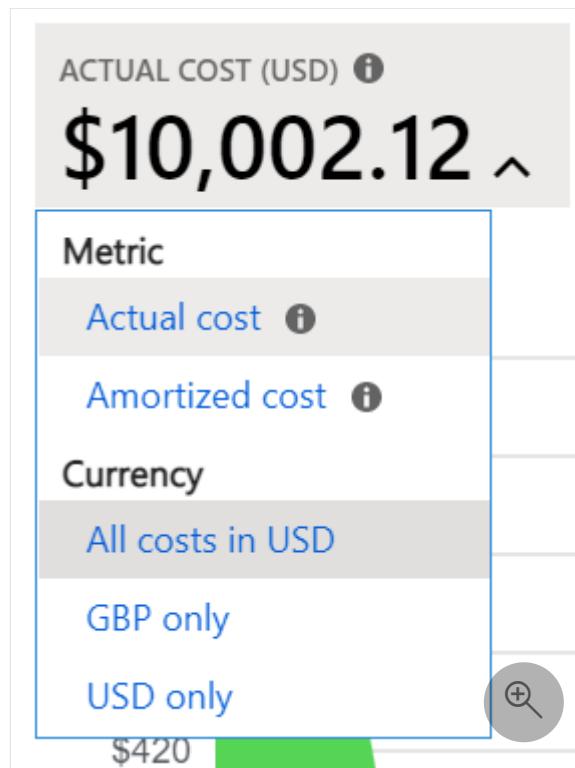
Amortized cost breaks down reservation purchases into daily chunks and spreads them over the duration of the reservation term. Most reservation terms are one or three years. Let's look at a one-year reservation example. Instead of seeing a \$365 purchase on January 1, you'll see a \$1.00 purchase every day from January 1 to December 31. In addition to basic amortization, these costs are also reallocated and associated by using the specific resources that used the reservation. For example, if that \$1.00 daily charge was split between two virtual machines, you'd see two \$0.50 charges for the day. If part of the reservation isn't utilized for the day, you'd see one \$0.50 charge associated with the applicable virtual machine and another \$0.50 charge with a charge type of UnusedReservation. Unused reservation costs can be seen only when you view amortized cost.

If you buy a one-year reservation on May 26 with an upfront payment, the amortized cost is divided by 365 (assuming it's not a leap year) and spread from May 26 through May 25 of the next year. If you pay monthly, the monthly fee is divided by the number of days in that month. The free is spread evenly across May 26 through June 25, with the next month's fee spread across June 26 through July 25.

Because of the change in how costs are represented, it's important to note that actual cost and amortized cost views will show different total numbers. In general, the total cost of months with a reservation purchase will decrease when you view amortized costs, and months following a reservation purchase will increase. Amortization is available only for reservation purchases and doesn't apply to Azure Marketplace purchases at this time.

Select a currency

Costs are shown in your billing currency by default. If you have charges in multiple currencies, costs will automatically be converted to USD. If you have any non-USD charges, you can switch between currencies in the total KPI menu. You may see options like **GBP only** to view only the charges in that one currency or **All costs in USD** to view the normalized costs in USD. You can't view costs normalized to other currencies today.



Select a budget

When you view a chart, it can be helpful to visualize your charges against a budget. It's especially helpful when showing accumulated daily costs with a forecast trending towards your budget. If your costs go over your budget, you'll see a red critical icon next to your budget. If your forecast goes over your budget, you'll see a yellow warning icon.

When you view daily or monthly costs, your budget may be estimated for the period. For instance, a monthly budget of \$31 will be shown as **\$1/day (est)**. Note your budget

won't be shown as red when it exceeds this estimated amount on a specific day or month.

Budgets that have filters aren't currently supported in cost analysis. You won't see them in the list. Budgets on lower-level scopes are also not shown in cost analysis today. To view a budget for a specific scope, change scope using the scope picker.

Change granularity

Use **Granularity** to indicate how you want to view cost over time. The lowest level you can view is Daily costs. You can view daily costs for up to 3 months or 92 consecutive days. If you select more than 92 days, cost analysis switches to **Monthly** granularity. It updates your date range to include the start and end of the selected months to provide the most accurate picture of your monthly costs. You can view up to 12 months of monthly costs.

If you'd like to view a running total of charges on either a daily or monthly basis, select **Accumulated**. Accumulated is especially helpful when you view your forecast as it helps you see the trend over time.

If you'd like to view the total for the entire period (no granularity), select **None**. Selecting no granularity is helpful when grouping costs by a specific attribute in either a chart or table.

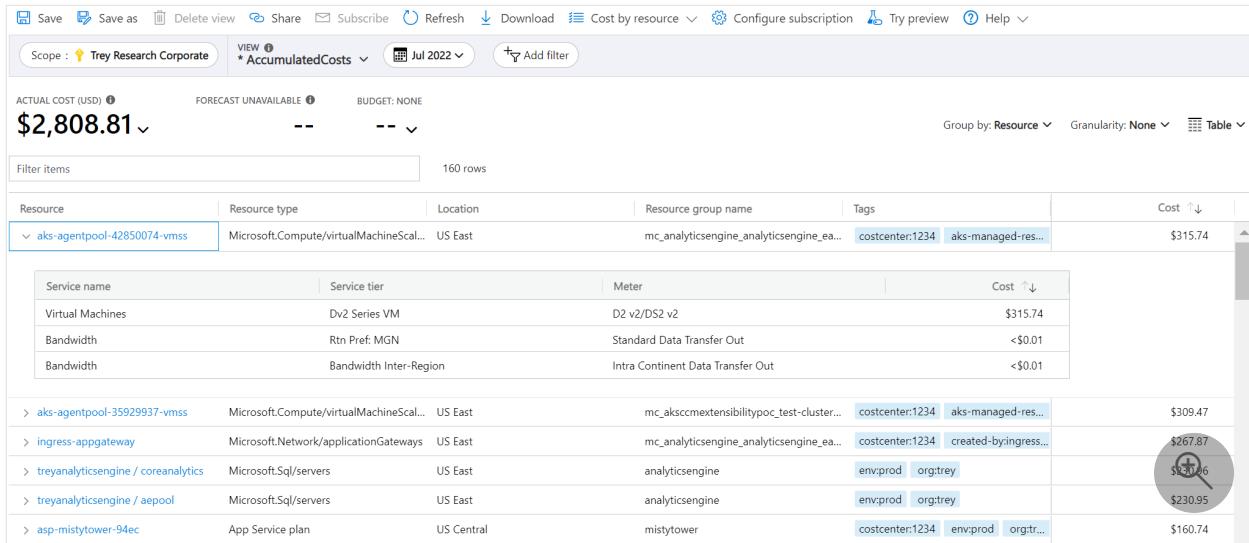
Visualize costs in a chart

Cost analysis supports the following chart types:

- Area charts are ideal for showing a running total with forecast trending towards a budget.
- Line charts are ideal for reviewing relative changes. Line charts aren't stacked, which helps spot changes easily.
- Column (stacked) charts are ideal for reviewing your daily or monthly run rate. It shows a breakdown by some attribute to easily spot which group has the most charges. Groups are sorted from largest to smallest from left-to-right, bottom-to-top.
- Column (grouped) charts are helpful when you view grouped costs with no granularity.

View costs in table format

You can view the full dataset for any view. Whichever selections or filters that you apply affect the data presented. To see the full dataset, select the **chart type** list and then select **Table** view.



The screenshot shows the Azure Cost Management portal with the following details:

- Scope:** Trey Research Corporate
- VIEW:** AccumulatedCosts
- Date:** Jul 2022
- Actual Cost (USD):** \$2,808.81
- Budget:** None
- Filter items:** 160 rows
- Group by:** Resource
- Granularity:** None
- Table View:** Selected

Resource	Resource type	Location	Resource group name	Tags	Cost ↑
aks-agentpool-42850074-vms	Microsoft.Compute/virtualMachineScal...	US East	mc_analyticsengine_analyticsengine_ea...	costcenter:1234 aks-managed-res...	\$315.74
Service name	Service tier	Meter		Cost ↓	
Virtual Machines	Dv2 Series VM	D2 v2/DS2 v2		\$315.74	
Bandwidth	Rtn Pref: MGN	Standard Data Transfer Out		<\$0.01	
Bandwidth	Bandwidth Inter-Region	Intra Continent Data Transfer Out		<\$0.01	
> aks-agentpool-35929937-vmss	Microsoft.Compute/virtualMachineScal...	US East	mc_aksccmextensibilitypoc_test-cluster...	costcenter:1234 aks-managed-res...	\$309.47
> ingress-appgateway	Microsoft.Network/applicationGateways	US East	mc_analyticsengine_analyticsengine_ea...	costcenter:1234 created-by:ingress...	\$267.87
> treyanalyticsengine / coreanalytics	Microsoft.Sql/servers	US East	analyticsengine	env:prod org:trey	\$210.46
> treyanalyticsengine / aepool	Microsoft.Sql/servers	US East	analyticsengine	env:prod org:trey	\$230.95
> asp-mistytower-94ec	App Service plan	US Central	mistytower	costcenter:1234 env:prod org:tr...	\$160.74

Next steps

- Learn about [Saving and sharing customized views](#).

Save and share customized views

Article • 03/10/2023

Cost analysis is used to explore costs and get quick answers for things like finding the top cost contributors. Or, understanding how you're charged for the services you use. As you analyze cost, you may find specific views you want to save or share with others.

Save and share cost views

A view is a saved query in Cost Management. When you save a view, all settings in cost analysis are saved, including filters, grouping, granularity, the main chart type, and donut charts. Underlying data isn't saved. Only you can see private views, while everyone with Cost Management Reader access or greater to the scope can see shared views.

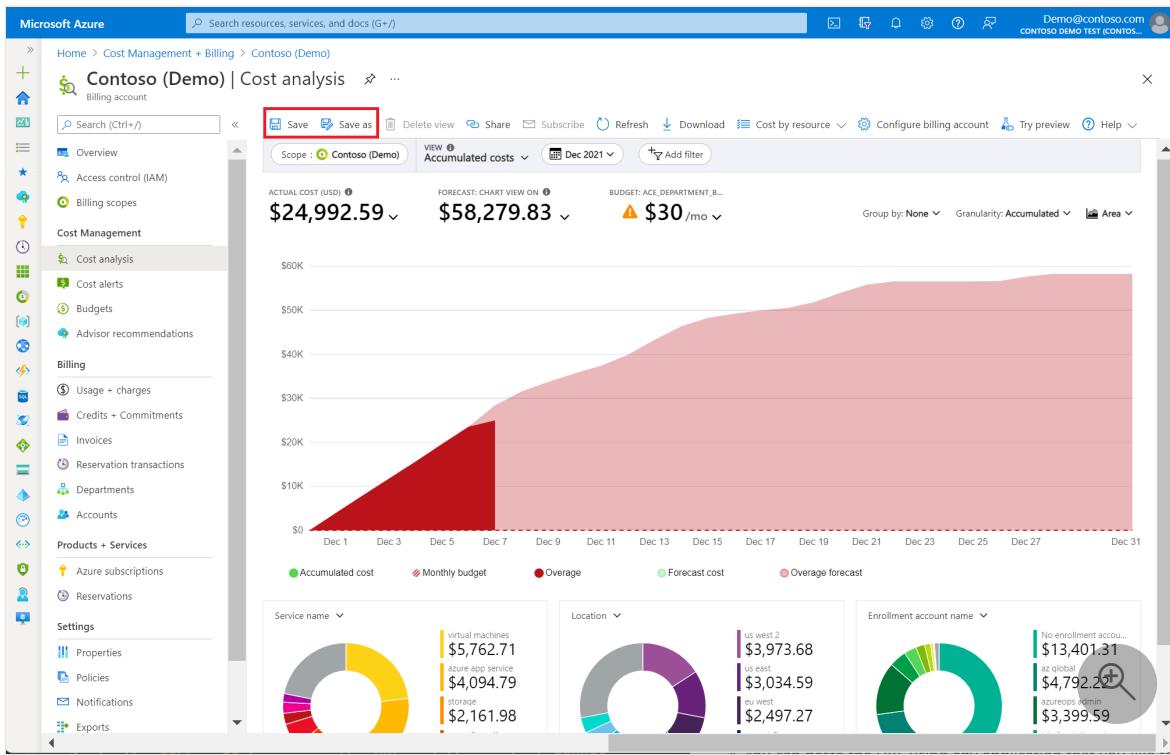
Check out the [Sharing and saving views](#) video.

After you save a view, you can share a link to it with others using the **Share** command. The link is specific to your current scope and view configuration. The link doesn't grant others access to the view itself, which may change over time, or the underlying data. If you don't have access to the scope, an `access denied` message is shown. We recommend using the Cost Management Contributor role to allow others to save and share views with others.

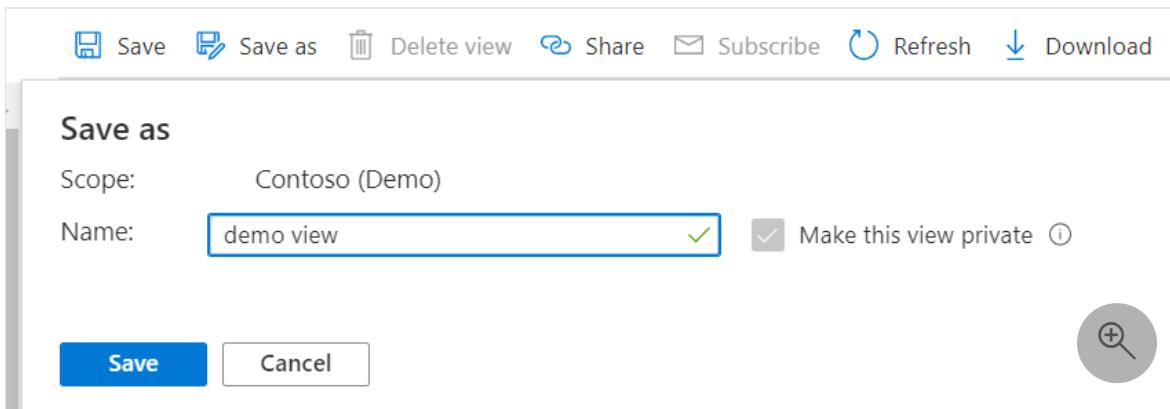
You can also pin the current view to an Azure portal dashboard. Pinning only includes a snapshot of the main chart or table and doesn't update when the view is updated. A pinned dashboard isn't the same thing as a saved view.

To save a view

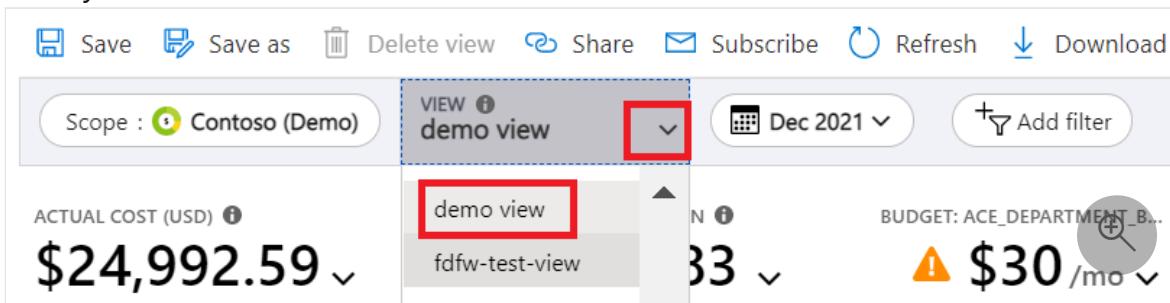
1. In cost analysis, make sure that the settings that you want saved are chosen.
2. Select the **Save** command at the top of the page to update your current view or **Save as** to save a new view.



3. Enter a name for the view and then select **Save**.



4. After you save a view, it's available to select from the View menu.

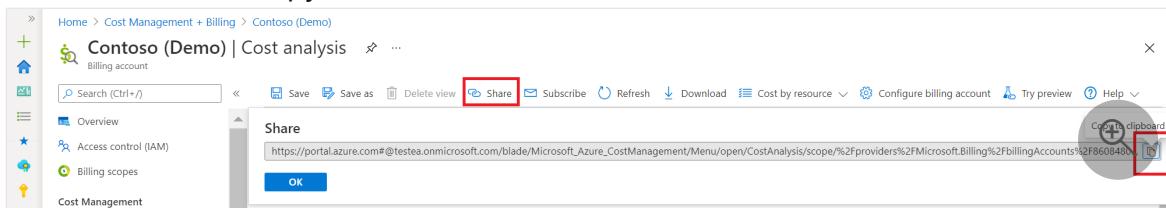


You can save up to 100 private views across all scopes for yourself and up to 100 shared views per scope that anyone with Cost Management Reader or greater access can use.

To share a view

1. In cost analysis, ensure that the currently selected view is the one that you want to share.
2. Select the **Share** command at the top of the page.

3. In the Share box, copy the URL and then select OK.



4. You can paste the URL using any application that you like to send to others.

If you need to generate a link to a view programmatically, use one of the following formats:

- View configuration – `https://<portal-domain>/@<directory-domain>/#blade/Microsoft_Azure_CostManagement/Menu/open/costanalysis/scope/<scope-id>/view/<view-config>`
- Saved view – `https://<portal-domain>/@<directory-domain>/#blade/Microsoft_Azure_CostManagement/Menu/open/costanalysis/scope/<scope-id>/viewId/<view-id>`

Use the following table for each property in the URL.

URL property	Description
portal-domain	Primary domain for the Azure portal. For example, <code>portal.azure.com</code> or <code>portal.azure.us</code>).
directory-domain	Domain used by your Azure Active Directory. You can also use the tenant ID. If it's omitted, the portal tries to use the default directory for the user that selected the link - it might differ from the scope.
scope-id	Full Resource Manager ID for the resource group, subscription, management group, or billing account you want to view cost for. If not specified, Cost Management uses the last view the user used in the Azure portal. The value must be URL encoded.
view-config	Encoded view configuration. See the following details. If not specified, cost analysis uses the <code>view-id</code> parameter. If neither are specified, cost analysis uses the built-in Accumulated cost view.
view-id	Full Resource Manager ID for the private or shared view to load. This value must be URL encoded. If not specified, cost analysis uses the <code>view</code> parameter. If neither are specified, cost analysis uses the built-in Accumulated cost view.

The `view-config` parameter is an encoded version of the JSON view configuration. For more information about the view body, see the [Views API reference](#). To learn how to build specific customizations, pin the desired view to an empty Azure portal dashboard, then download the dashboard JSON to review the JSON view configuration.

After you have the desired view configuration:

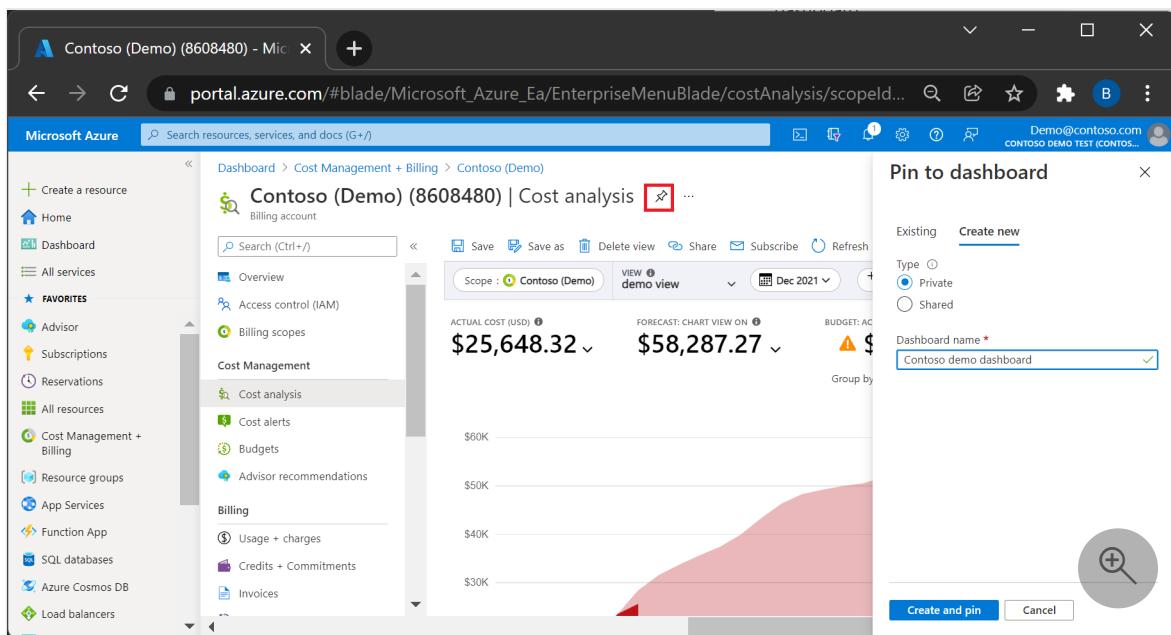
1. Use Base 64 encode for the JSON view configuration.
2. Use Gzip to compress the encoded string.
3. URL-encode the compressed string.
4. Add the final encoded string to the URL after the `/view/` parameter.

Pin a view to the Azure portal dashboard

As mentioned previously, pinning a view to an Azure portal dashboard only saves the main chart or table. It's essentially a thumbnail you can select to get back to the view configuration in cost analysis. Keep in mind the dashboard tile is a copy of your view configuration – if you save a view that was previously pinned, the pinned tile doesn't update. To update the tile, pin the saved view again.

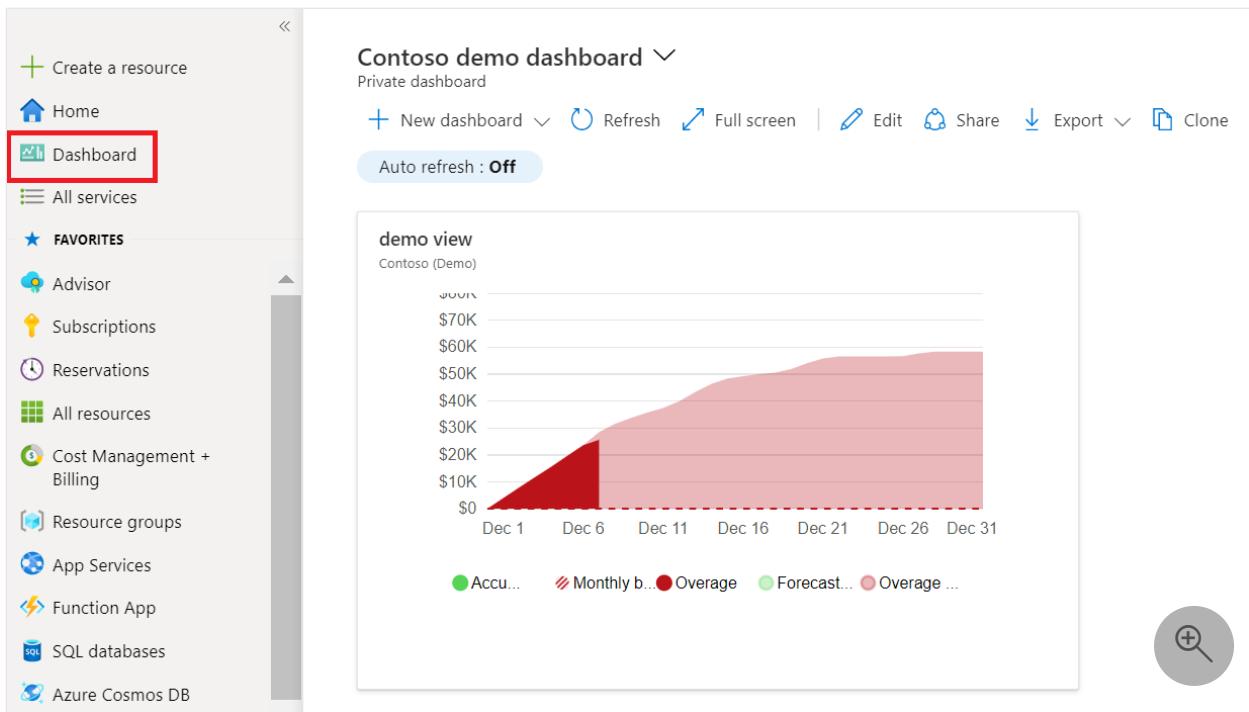
To pin cost analysis to a dashboard

1. In cost analysis, ensure that the currently selected view is the one that you want to pin.
2. To the right of your billing scope or subscription name, select the **Pin** symbol.
3. In the Pin to dashboard window, choose **Existing** to pin the current view to the existing dashboard or choose **Create new** to pin the current view to a new dashboard.



4. Select **Private** to if you don't want to share the dashboard and then select **Pin** or select **Shared** to share the dashboard with others and then select **Pin**.

To view the dashboard after you've pinned it, from the Azure portal menu, select **Dashboard**.



To rename a tile

1. From the dashboard where your tile is pinned, select the title of the tile you want to rename. This action opens cost analysis with that view.
2. Select the **Save** command at the top of the page.
3. Enter the name of the tile you want to use.
4. Select **Save**.
5. Select the **Pin** symbol to the right of the page header.
6. From the dashboard, you can now remove the original tile.

For more advanced dashboard customizations, you can also export the dashboard, customize the dashboard JSON, and upload a new dashboard. Dashboard creations can include other tile sizes or names without saving new views. For more information, see [Create a dashboard in the Azure portal](#).

Download data or charts

When you want to share information with others that don't have access to the scope, you can download the view in PNG, Excel, and CSV formats. Then you can share it with them by email or other means. The downloaded data is a snapshot, so it isn't automatically updated.

The screenshot shows the Azure Cost Management + Billing portal. In the top navigation bar, it says 'Contoso (Demo) (8608480) | Cost analysis'. Below the navigation, there's a search bar and several action buttons: Save, Save as, Delete view, Share, Subscribe, Refresh, and Download (which is highlighted with a red box). The scope is set to 'Contoso (Demo)'. The view is 'Accumulated costs' for the month of Dec 2021. A large chart at the bottom shows a comparison between actual costs and forecasts across different categories.

When you download data, cost analysis includes summarized data as it's shown in the table. The cost by resource view includes all resource meters in addition to the resource details. If you want a download of only resources and not the nested meters, use the cost analysis preview. You can access the preview from the **Cost by resource** menu at the top of the page, where you can select the Resources, Resource groups, Subscriptions, Services, or Reservations view.

If you need more advanced summaries or you're interested in raw data that hasn't been summarized, schedule an export to publish raw data to a storage account on a recurring basis.

Subscribe to scheduled alerts

In addition to saving and opening views repeatedly or sharing them with others manually, you can also subscribe to updates or a recurring schedule to get alerted as costs change. You can also set up alerts to be shared with others who may not have direct access to costs in the portal.

To subscribe to scheduled alerts

1. In Cost analysis, select any chart view you want to subscribe to or create and save a new chart view.
 - Built-in views (for example, Accumulated costs, Daily costs, or Cost by service) can't be changed. If you need to change the date range, currency, amortization, or any other setting, you need to save that as a private or shared view.
2. Select **Subscribe** at the top of the page.
3. Select **+ Add** at the top of the list of alerts.
4. Specify the desired email settings and select **Save**.
 - The **Name** helps you distinguish the different emails setup for the current view. Use it to indicate audience or purpose of this specific email.
 - The **Subject** is what people see when they receive the email.

- You can include up to 20 recipients. Consider using a distribution list if you have a large audience. To see how the email looks, start by sending it only to yourself. You can update it later.
- The **Message** is shown in the email to give people more context about why they're receiving the email. You may want to include what it covers, who requested it, or who to contact to make changes.
- To share data with people who don't have access to the scope or view, select **Add a CSV download link** to include an unauthenticated link to the data.
- To allow people with write access to the scope to modify the email configuration settings, clear the **Make this schedule private** option. Doing so allows billing account admins or Cost Management Contributors to edit the alert, in addition to the person who created it. By default, the option is selected, meaning that only the creator of the scheduled alert can see or edit it.
- The **Start date** is when you start receiving the email. It defaults to the current day.
- The **End date** is when you receive the last email. It can be up to one year from the current day, which is the default. You can update it later.
- The **Frequency** indicates how often you want the email to be sent. It's based on the start date, so if you want a weekly email on a different day of the week, change the start date first. To get an email after the month is closed, select **After invoice finalized**. Ensure your view is looking at last month. If you use the current month, it only sends you the first few days of the month. By default, all emails are sent at 8:00 AM local time. To customize any of the options, select **Custom**.

 **Note**

Scheduled alerts are set to your local time. However emails are sent at a fixed time in the UTC time zone. Daylight savings time isn't accounted for, which might result in a one-hour variation in your actual delivery time.

5. After you save the alert, a list of configured alerts for the current view is shown. If you want to see a preview of the email, select the row and select **Send now** at the top to send the email to all recipients.

Keep in mind that if you choose to include a link to data, anyone who receives the email has access to the data included in that email. Data expires after seven days.

Frequently asked questions

The following sections cover the most commonly asked questions and answers about saving and sharing customized views in Cost Management.

Why am I not receiving emails from scheduled alerts?

There could be a few reasons why you're not receiving alert emails. Try the following actions:

- Confirm that your email address is shown as a recipient and that it was entered correctly.
- Check your spam or junk mail folder for emails from `microsoft-noreply@microsoft.com`.
- Check to see if the alert is expired, disabled, or deleted. You can extend, reenable, or create a new scheduled alert to fix the problem.
- Work with your admin to reenable the [view charges policy](#) in the Azure portal. The policy applies to indirect Enterprise Agreements and to Microsoft Customer Agreements with a Microsoft partner.

Why can't I use the Subscribe command in cost analysis?

The **Subscribe** command is currently unavailable for two scenarios in cost analysis:

- When using a management group scope
- When working with table views

How far in the future can I set the end date?

The end date can be anywhere from one day to one year from today or the start date, whichever is later. You can change or extend it to one year from the current day until the schedule expires. This limitation is in place to ensure that alert subscriptions remain relevant and accurate.

For example, if you create a scheduled alert on March 3, 2023, the end date can be any date from March 4, 2023, to March 3, 2024.

It's important to note that you can edit the end date of an existing scheduled alert at any time. If the end date has already passed and you want to continue receiving the alert, you must edit the scheduled alert with a future end date.

When do I receive alert emails?

You can choose when emails arrive by setting the **Frequency** to **Custom**. You should receive the email before that time or within an hour of that time. If there are unforeseen issues, emails could be delayed for up to a day.

Although the configuration of scheduled alerts considers your local time, the actual sending of emails happens at the same time in the UTC time zone. The time conversion doesn't account for daylight savings time. It might result in a one-hour variation in your actual delivery time.

Next steps

- For more information about creating dashboards, see [Create a dashboard in the Azure portal](#).
- To learn more about Cost Management, see [Cost Management + Billing documentation](#).

Identify anomalies and unexpected changes in cost

Article • 04/05/2023

The article helps you identify anomalies and unexpected changes in your cloud costs using Cost Management and Billing. You start with anomaly detection for subscriptions in cost analysis to identify any atypical usage patterns based on your cost and usage trends. Then, you learn how to drill into cost information to find and investigate cost spikes and dips.

You can also create an anomaly alert to automatically get notified when an anomaly is detected.

In general, there are three types of changes that you might want to investigate:

- New costs—For example, a resource that was started or added such as a virtual machine. New costs often appear as a cost starting from zero.
- Removed costs—For example, a resource that was stopped or deleted. Removed costs often appear as costs ending in zero.
- Changed costs (increased or decreased)—For example, a resource was changed in some way that caused a cost increase or decrease. Some changes, like resizing a virtual machine, might be surfaced as a new meter that replaces a removed meter, both under the same resource.

Identify cost anomalies

The cloud comes with the promise of significant cost savings compared to on-premises costs. However, savings require diligence to proactively plan, govern, and monitor your cloud solutions. Even with proactive processes, cost surprises can still happen. For example, you might notice that something has changed, but you're not sure what. Using Cost Management anomaly detection for your subscriptions can help minimize surprises.

Whether you know if you have any existing cost anomalies or not, Cost analysis informs you if it finds anything unusual as part of Insights. If not, Cost analysis shows **No anomalies detected**.

View anomalies in Cost analysis

Anomaly detection is available in Cost analysis (preview) when you select a subscription scope. You view your anomaly status as part of **Insights**. And as with other insights ↗, the experience is simple.

In the Azure portal, navigate to Cost Management from Azure Home. Select a subscription scope and then in the left menu, select **Cost analysis**. In the view list, select any view under **Preview views**. In the following example, the **Resources** preview view is selected. If you have a cost anomaly, you see an insight.

The screenshot shows the Azure Cost Management interface for the 'Trey Research Corporate' subscription. The main view is the 'Resources' preview. The total cost for the month is displayed as \$11,280. An insight is shown for a daily run rate of 1748% on September 28, with a link to 'See insights'. The left sidebar includes sections for Cost analysis (selected), Cost alerts, Budgets, Advisor recommendations, Billing, Products + services, Settings, and Support > troubleshooting.

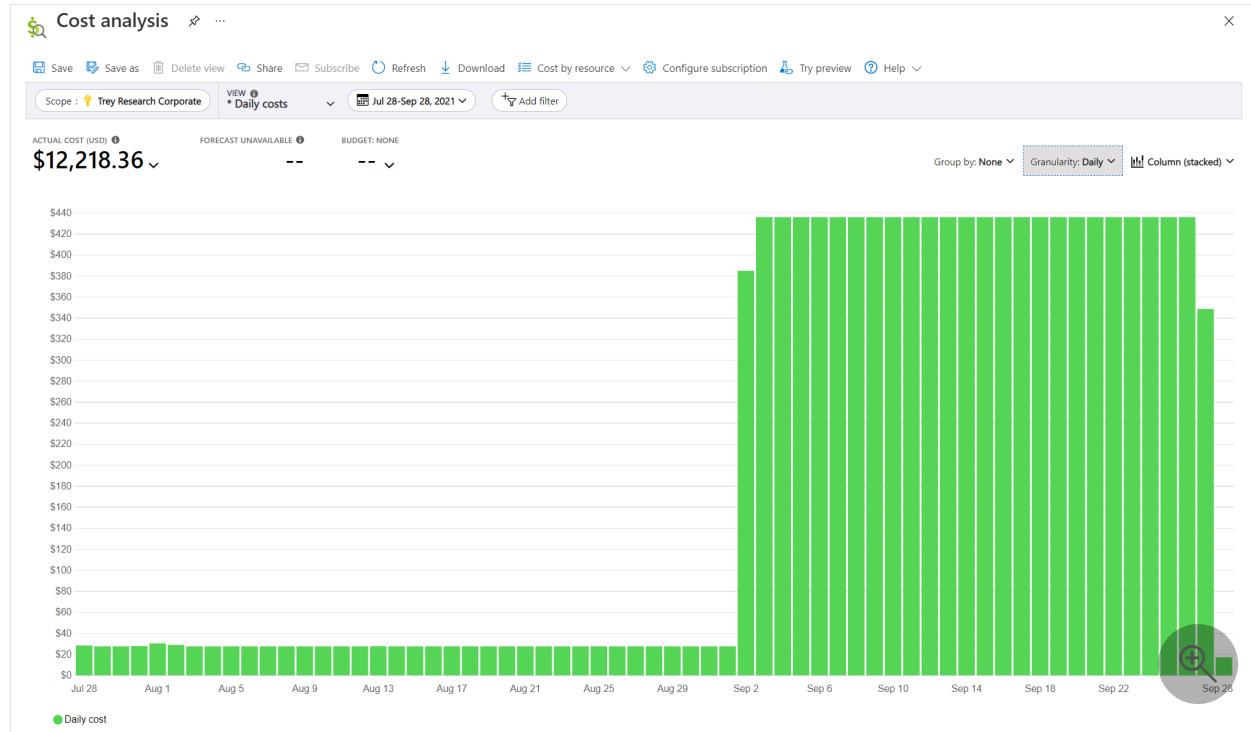
If you don't have any anomalies, you see a **No anomalies detected** insight, confirming the dates that were evaluated.

The screenshot shows the 'Resource groups' preview view. It displays 8 resource groups: analyticsengine, mistytower, and mc_analyticsengine_analyticsengine_eastus. The 'Insights' section indicates 'No anomalies detected' for the period of October 1-5, stating that daily cost was within the expected range based on average daily cost over the last 60 days. An 'Is this helpful?' button with a magnifying glass icon is also present.

Drill into anomaly details

To drill into the underlying data for something that has changed, select the insight link. It opens a view in classic cost analysis where you can review your daily usage by resource group for the time range that was evaluated.

Continuing from the previous example of the anomaly labeled **Daily run rate down 748% on Sep 28**, let's examine its details after the link is selected. The following example image shows details about the anomaly. Notice the large increase in costs, a cost spike, and eventual drop in from a temporary, short-lived resource.



Cost anomalies are evaluated for subscriptions daily and compare the day's total usage to a forecasted total based on the last 60 days to account for common patterns in your recent usage. For example, spikes every Monday. Anomaly detection runs 36 hours after the end of the day (UTC) to ensure a complete data set is available.

The anomaly detection model is a univariate time-series, unsupervised prediction and reconstruction-based model that uses 60 days of historical usage for training, then forecasts expected usage for the day. Anomaly detection forecasting uses a deep learning algorithm called [WaveNet](#). It's different than the Cost Management forecast. The total normalized usage is determined to be anomalous if it falls outside the expected range based on a predetermined confidence interval.

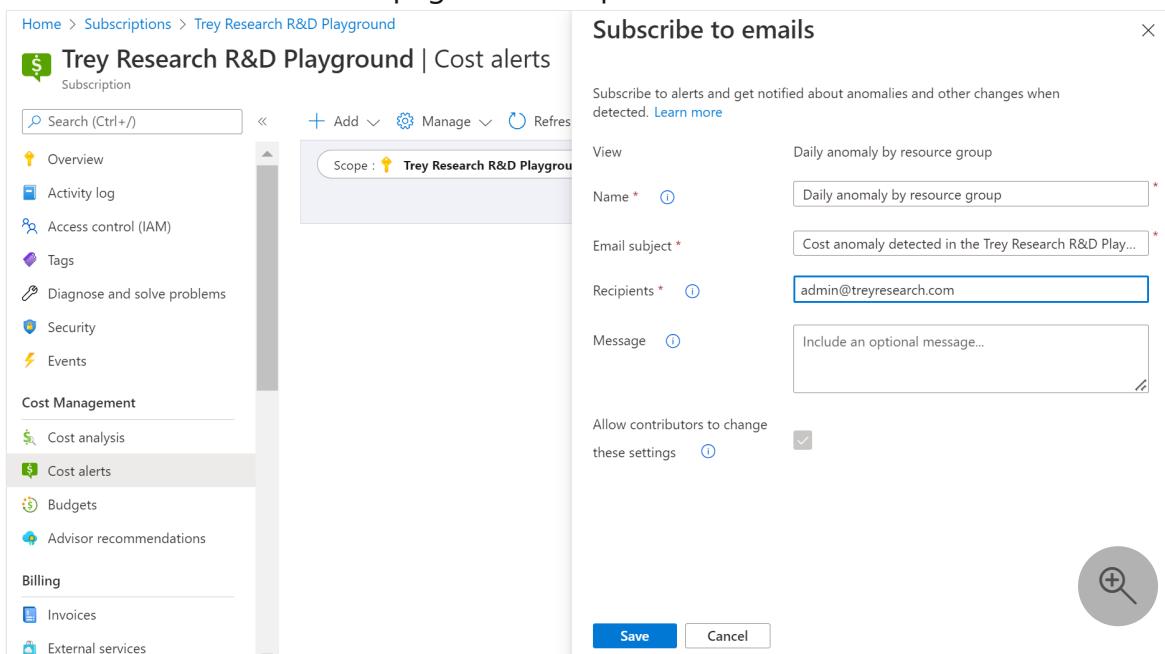
Anomaly detection is available to every subscription monitored using the cost analysis preview. To enable anomaly detection for your subscriptions, open the cost analysis preview and select your subscription from the scope selector at the top of the page. You see a notification informing you that your subscription is onboarded and you start to see your anomaly detection status within 24 hours.

Create an anomaly alert

You can create an alert to automatically get notified when an anomaly is detected. Creating an anomaly alert requires the Cost Management Contributor or greater role or the `Microsoft.CostManagement/scheduledActions/write` permission for custom roles. For more information, see [Feature behavior for each role](#).

An anomaly alert email includes a summary of changes in resource group count and cost. It also includes the top resource group changes for the day compared to the previous 60 days. And, it has a direct link to the Azure portal so that you can review the cost and investigate further.

1. From Azure Home, select **Cost Management** under Tools.
2. Verify you've selected the correct subscription in the scope at the top of the page.
3. In the left menu, select **Cost alerts**.
4. On the Cost alerts page, select **+ Add > Add anomaly alert**.
5. On the Subscribe to emails page, enter required information and then select **Save**.



The screenshot shows two overlapping windows. The background window is titled 'Trey Research R&D Playground | Cost alerts' and displays a sidebar with links like Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Security, Events, Cost analysis, Cost alerts (which is selected and highlighted in grey), Budgets, Advisor recommendations, Invoices, and External services. The foreground window is a modal titled 'Subscribe to emails' with the following fields:

- Name:** Daily anomaly by resource group
- Email subject:** Cost anomaly detected in the Trey Research R&D Play...
- Recipients:** admin@treyresearch.com
- Message:** (Text area placeholder: Include an optional message...)
- Allow contributors to change these settings:** A checked checkbox.

At the bottom of the modal are 'Save' and 'Cancel' buttons, and a large circular button with a plus sign.

Here's an example email generated for an anomaly alert.

Anomaly detected in Trey Research Corporate subscription



Microsoft Azure

Reply Reply All Forward

Mon 3/14/2022 3:37 PM

You forwarded this message on 3/14/2022 3:36 PM.
If there are problems with how this message is displayed, click here to view it in a web browser.



An unusual cost increase was detected on March 12, 2022 for the Trey Research Corporate subscription

Azure Cost Management and Billing detected a possible cost anomaly based on daily usage trends between January 15, 2021 and March 12, 2022. Please review changes to determine whether this was expected.

Subscription summary

Anomaly detected	Yes
Delta	10.12 %

Resource Groups summary

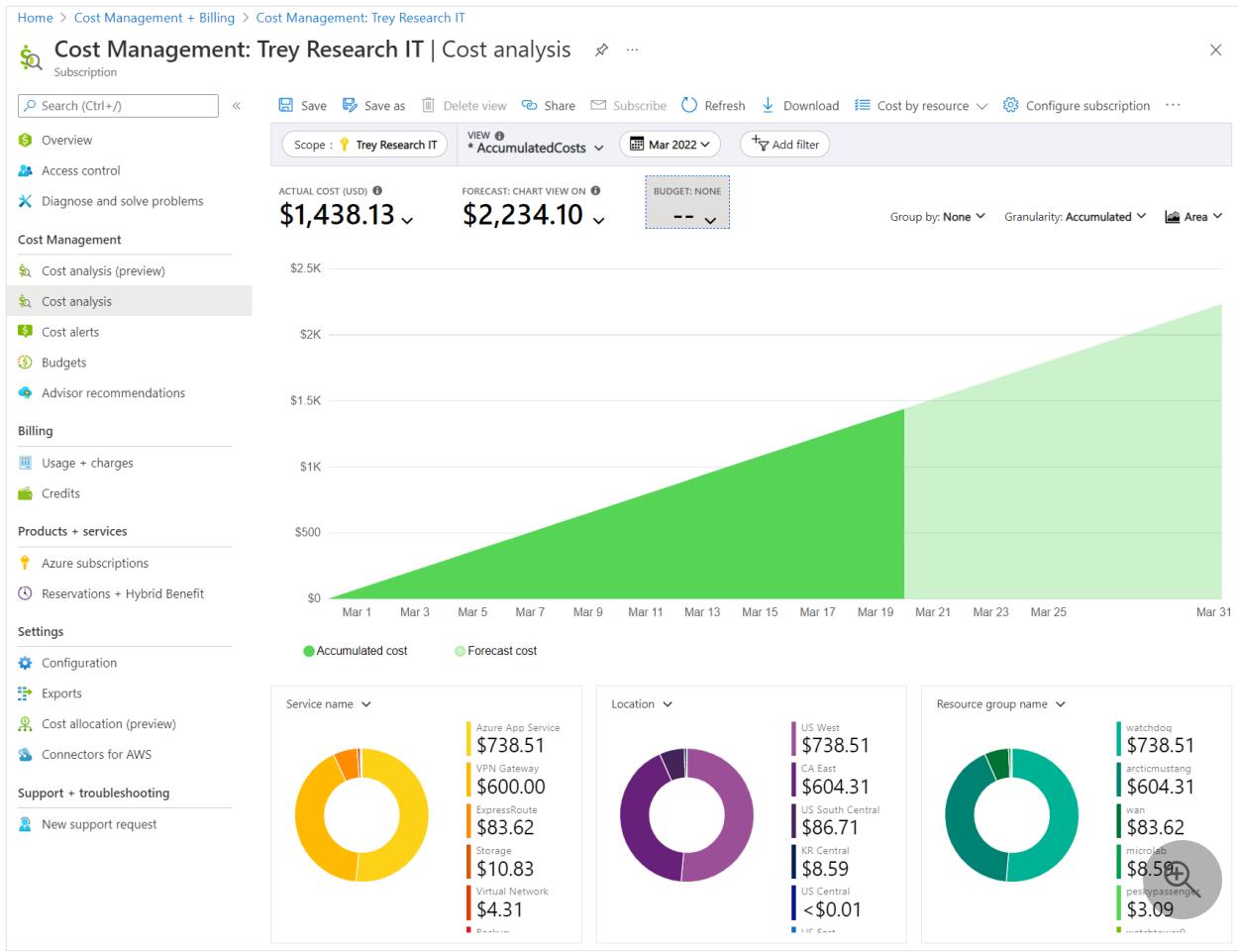
	Count	Cost change %
New	0	0 %
Existing	28	7.94 %
Removed ¹	0	0 %
Total ²	28	0 %

¹ "Removed" means there was no cost or usage generated. This may be due to stopped, moved, or deleted resources.



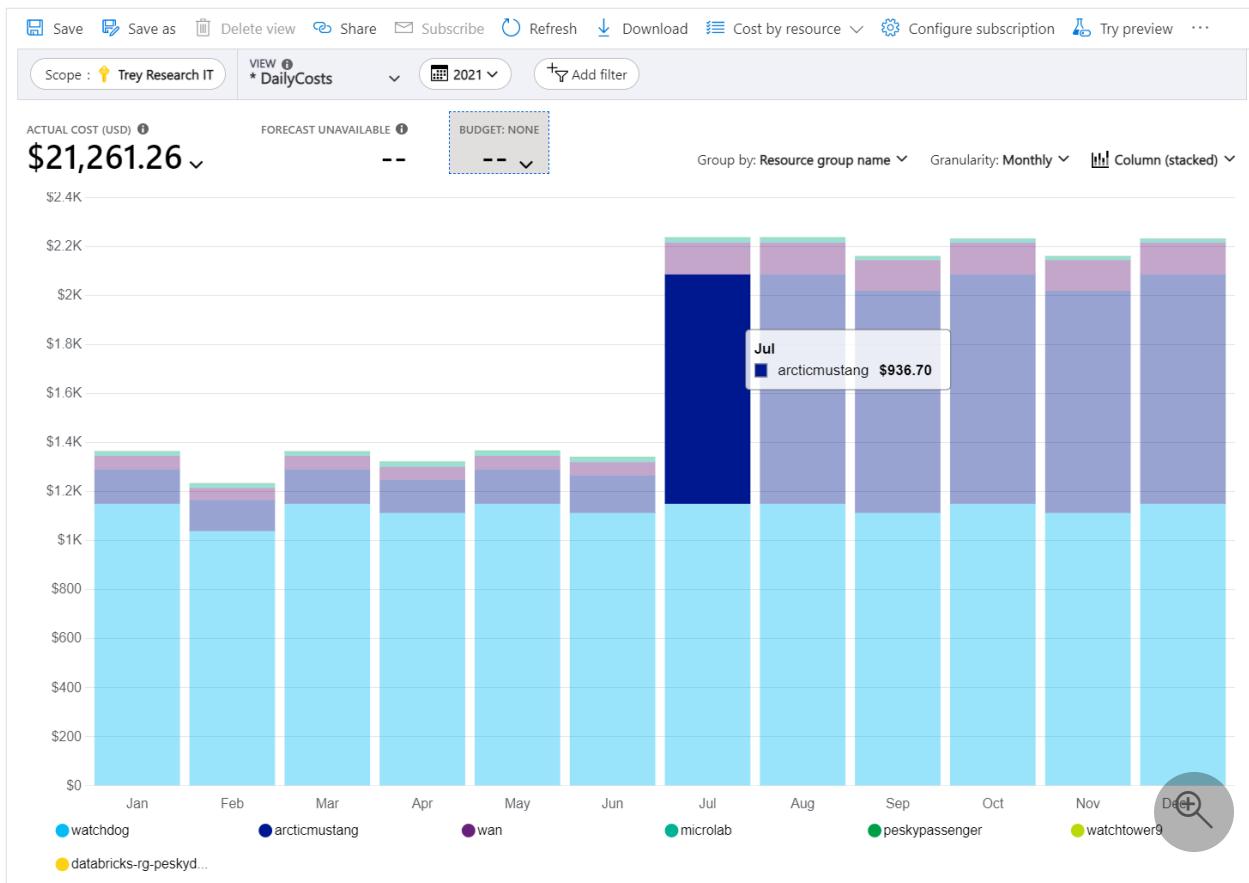
Manually find unexpected cost changes

Let's look at a more detailed example of finding a change in cost. When you navigate to Cost analysis and then select a subscription scope, you start with the **Accumulated costs** view. The following screenshot shows an example of what you might see.

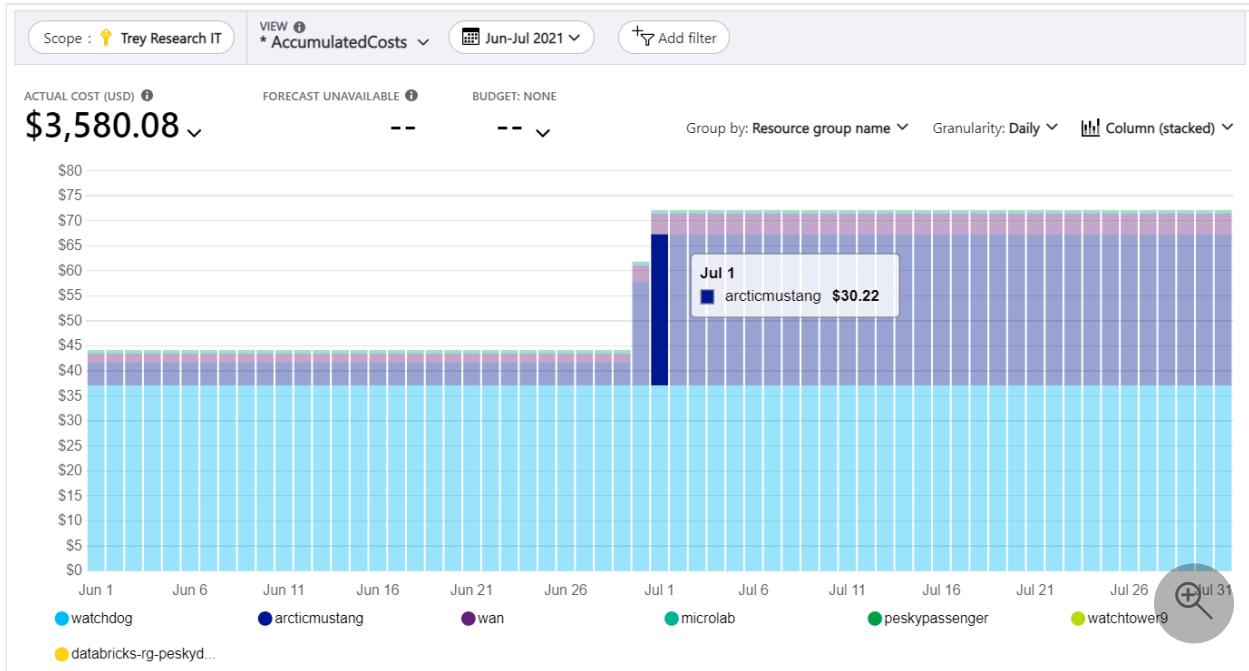


With the default view and current month (March 2022), the example image doesn't show any dips or spikes.

Change the view to **Daily costs** and then expand the date range to **Last year (2021)**. Then, set the granularity to **Monthly**. In the following image, notice that there's a significant increase in costs for the `arcticmustang` resource group starting in July.



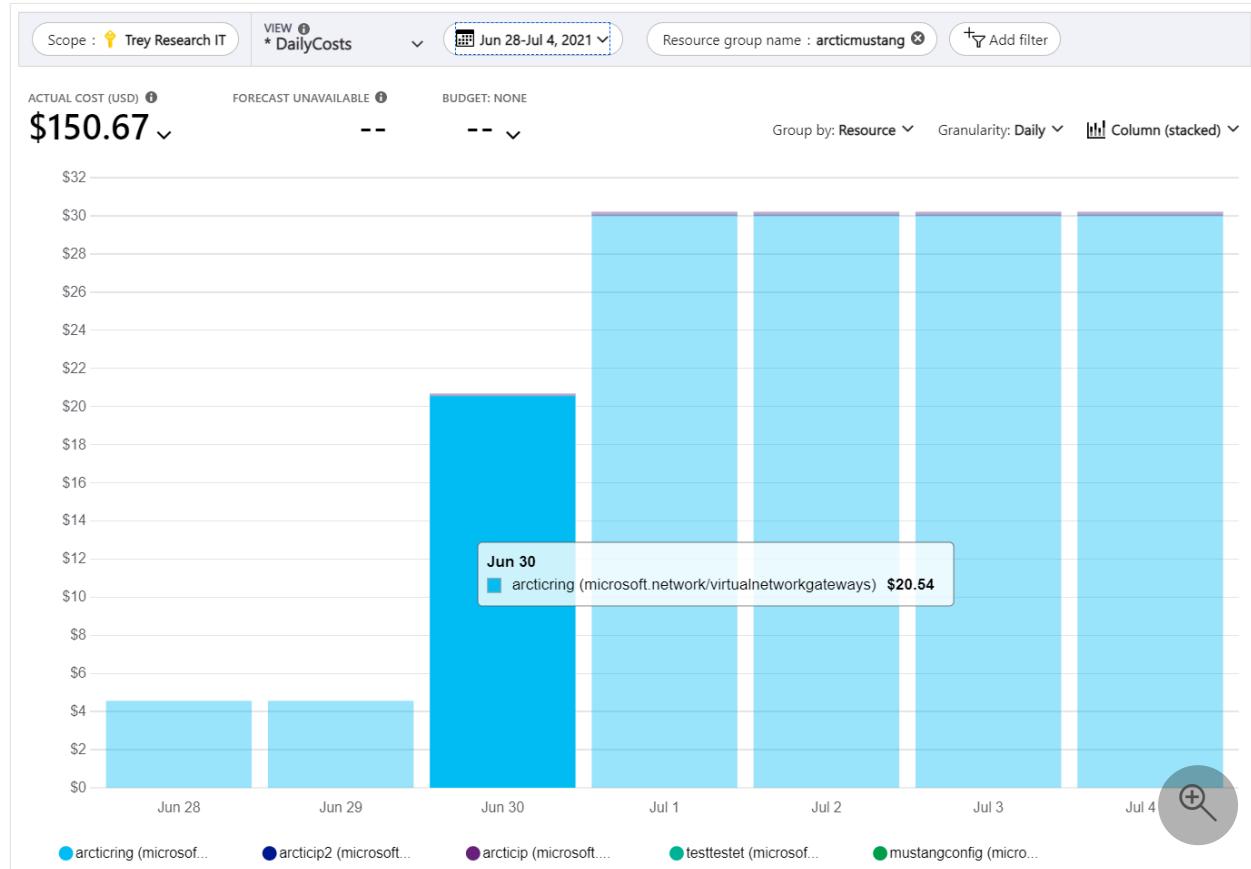
Let's examine the increase in cost for the resource group more fully. To drill into the time frame of the change, change the date range. In the following example, we set a custom date range from June to July 2021 and then set the Granularity to **Daily**. In the example, the daily cost for the resource group was about \$4.56. On June 30, the cost increased to \$20.68. Later on July 1 and after, the daily cost went to \$30.22.



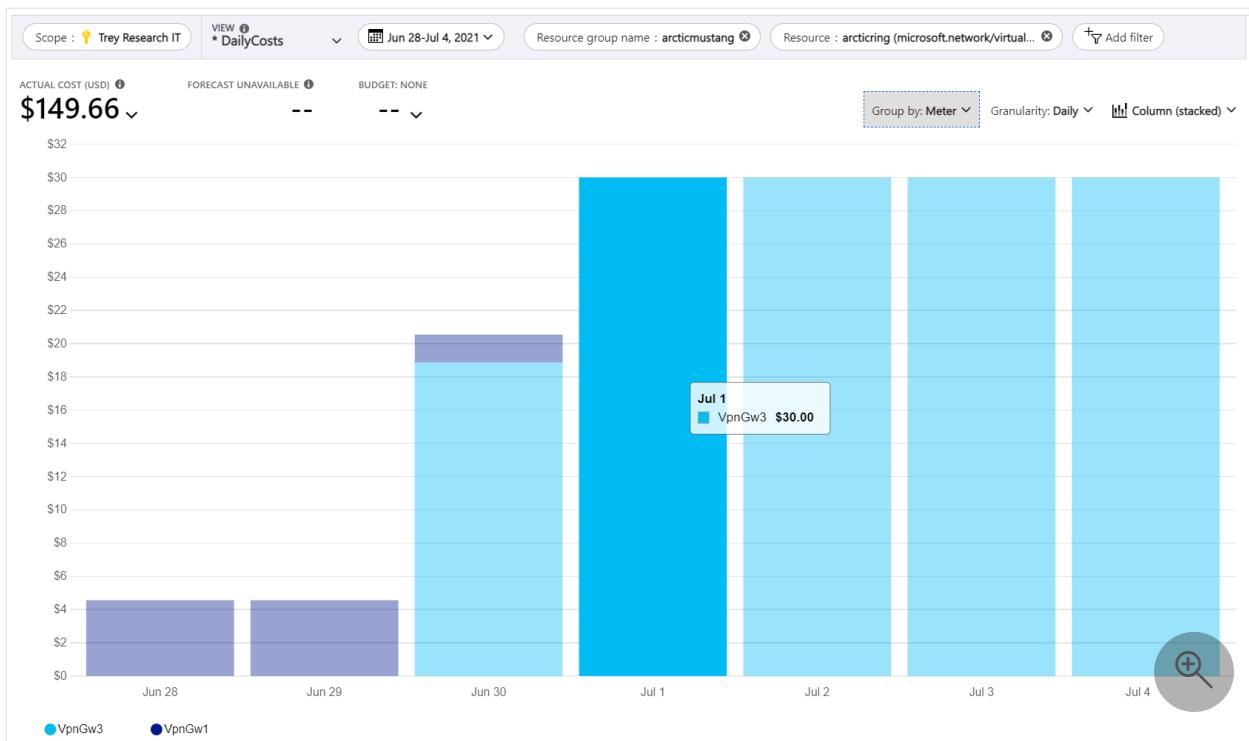
So far, we've found an increase in cost for the `arcticmustang` resource group at the end of June and the beginning of July. You might notice that the cost increase spanned over

two days. The change took two days because a change in the middle of a day doesn't show the full effect of that change until the following full day.

Let's continue drilling into the data to find out more about the cost increase. Select the item that increased in cost (`articmustang`) to automatically set a filter for the resource group name. Then, change the **Group by** list to **Resource**. Then set the date range to a smaller period. For example, June 28 to July 4. In the following example image, the increase in cost is clearly shown. The type of resource is shown as `microsoft.network/virtualnetworkgateways`.



Next, select the resource in the chart that increased in cost `articring` to set another filter for the resource. Now, costs are shown for just that resource. Then, set the **Group by** list to **Meter**.



In the previous example, you see that the virtual private network resource named VpnGw1 stopped getting used on June 30. On June 30, a more expensive virtual private network resource named VpnGw3 started getting used.

At this point, you know what changed and the value that costs changed. However, you might not know *why* the change happened. At this point, you should contact the people that created or used the resource. Continue to the next section to learn more.

Find people responsible for changed resource use

Using Cost analysis, you might have found resources that had sudden changes in usage. However, it might not be obvious who is responsible for the resource or why the change was made. Often, the team responsible for a given resource knows about changes that were made to a resource. Engaging them is useful as you identify why charges might appear. For example, the owning team may have recently created the resource, updated its SKU (thereby changing the resource rate), or increased the load on the resource due to code changes.

The [Get resource changes](#) article for Azure Resource Graph might help you to find additional information about configuration changes to resources.

Continue reading the following sections for more techniques to determine who owns a resource.

Analyze the audit logs for the resource

If you have permission to view a resource, you should be able to access its audit logs. Review the logs to find the user who was responsible for the most recent changes to a resource. To learn more, see [View and retrieve Azure Activity log events](#).

Analyze user permissions to the resource's parent scope

People that have write access to a subscription or resource group typically have information about the resources that were created or updated. They should be able to explain the purpose of a resource or point you to the person who knows. To identify the people with permissions for a subscription scope, see [Check access for a user to Azure resources](#). You can use a similar process for billing scopes, resource groups, and management groups.

Examine tagged resources

If you have an existing policy of [tagging resources](#), the resource might be tagged with identifying information. For example, resources might be tagged with owner, cost center, or development environment information. If you don't already have a resource tagging policy in place, consider adopting one to help identify resources in the future.

Get help to identify charges

If you've used the preceding strategies and you still don't understand why you received a charge or if you need other help with billing issues, [create a support request](#) ↗.

Next steps

- Learn about how to [Optimize your cloud investment with Cost Management](#).

Common cost analysis uses

Article • 06/15/2023

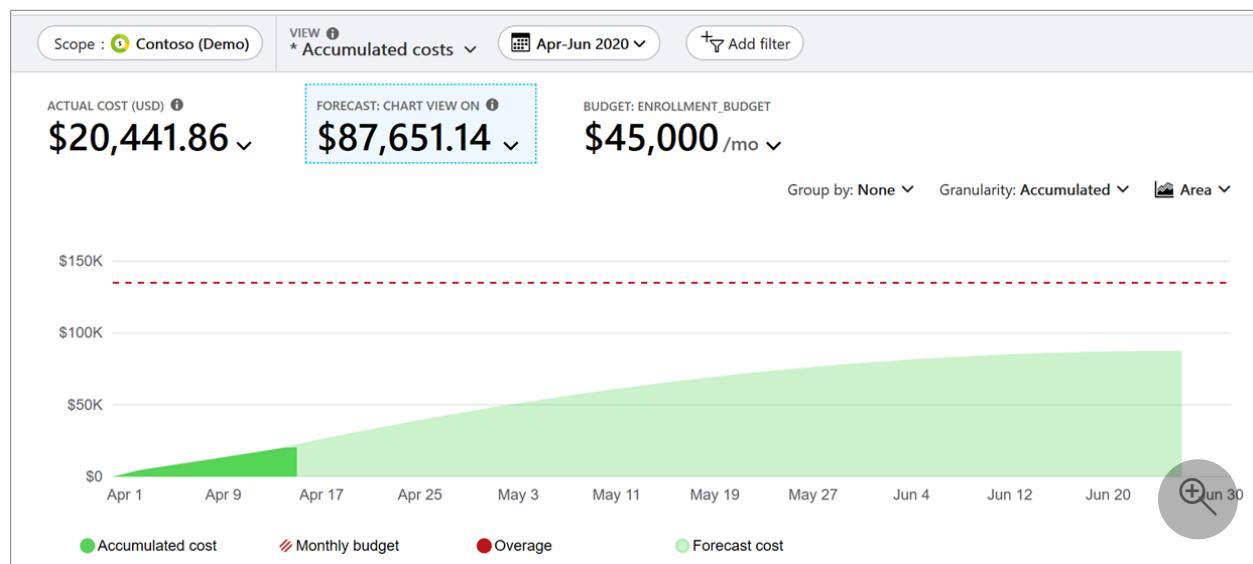
Cost Management users often want answers to questions that many others ask. This article walks you through getting results for common cost analysis tasks in Cost Management.

View forecast costs

Forecast costs are shown in cost analysis areas for area and stacked column views. The forecast is based on your historical resource use. Changes to your resource use affect forecast costs.

In the Azure portal, navigate to cost analysis for your scope. For example: **Cost Management + Billing > Cost Management > Cost analysis**.

In the default view, the top chart has the Actual/Amortized cost and forecast cost sections. The solid color of the chart shows your Actual/Amortized cost. The shaded color shows the forecast cost.



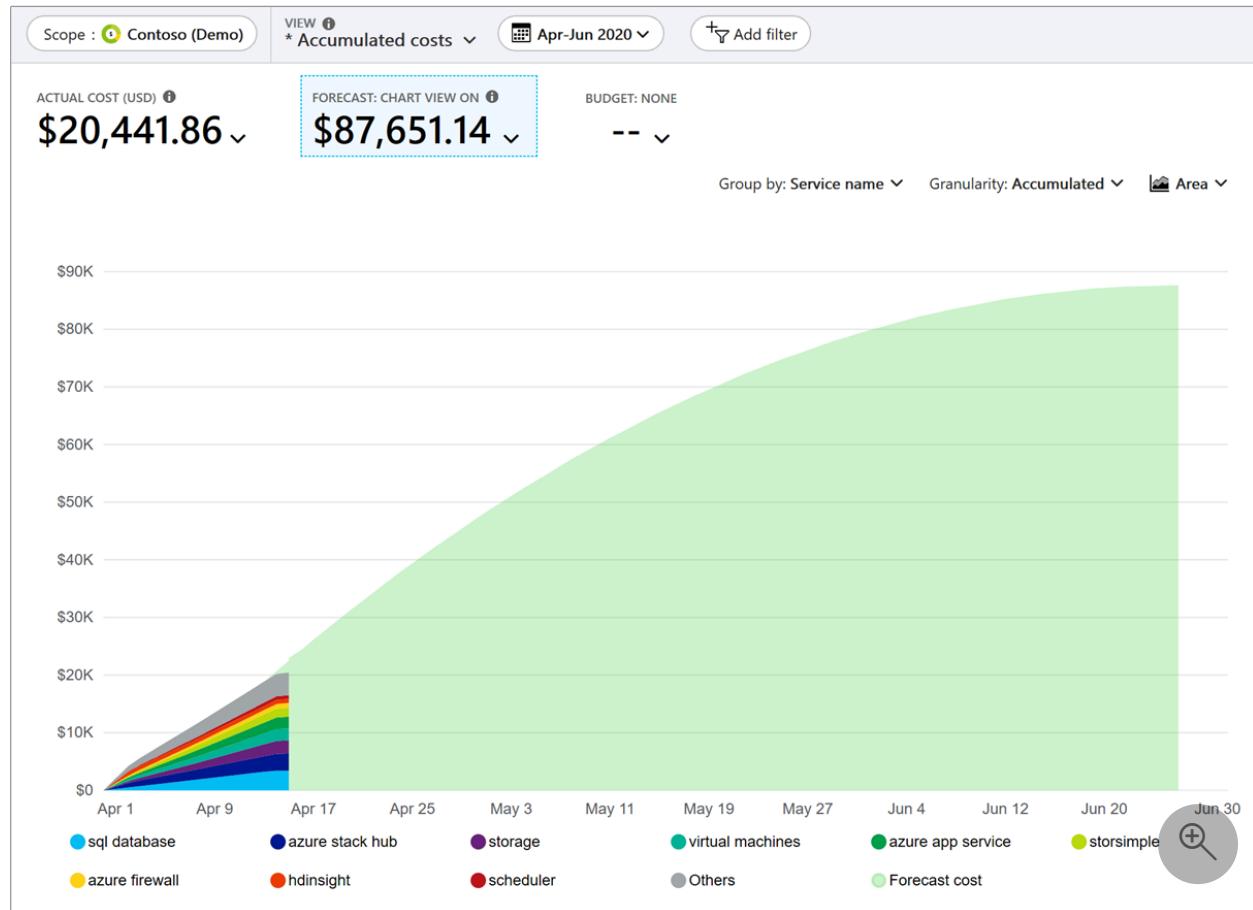
View forecast costs grouped by service

The default view doesn't show forecast costs group by a service, so you have to add a group by selection.

In the Azure portal, navigate to cost analysis for your scope. For example: **Cost Management + Billing > Cost Management > Cost analysis**.

Select **Group by > Service name**.

The view shows your costs grouped for each service. The forecast cost isn't calculated for each service. It's projected for the **Total** of all your services.



View forecast costs for a service

You can view forecast costs narrowed to a single service. For example, you might want to see forecast costs for just virtual machines.

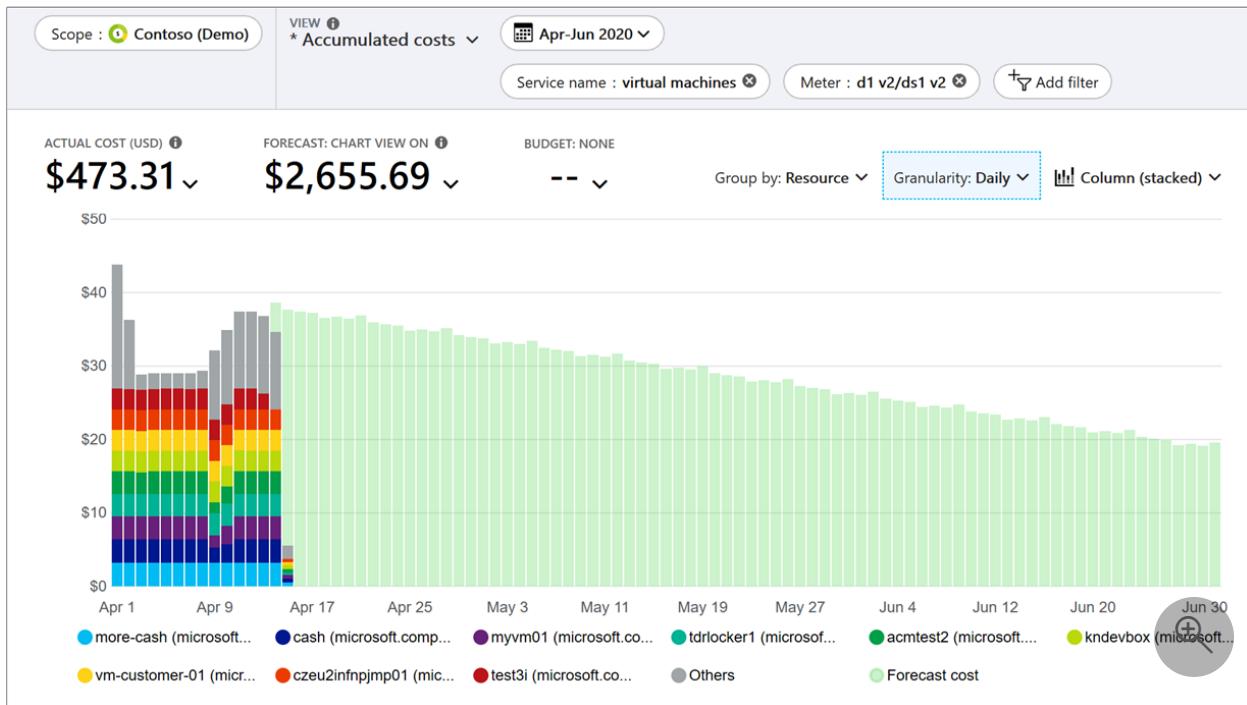
1. In the Azure portal, navigate to cost analysis for your scope. For example: [Cost Management + Billing > Cost Management > Cost analysis](#).
2. Select **Add filter** and then select **Service name**.
3. In the **choose** list, select a service. For example select, **virtual machines**.

Review the actual cost for selection and the forecast cost.

You can add more customizations to the view.

1. Add a second filter for **Meter** and select a value to filter for an individual type of meter under your selected service name.
2. Group by **Resource** to see the specific resources that are accruing cost. The forecast cost isn't calculated for each service. It's projected for the **Total** of all your

resources.



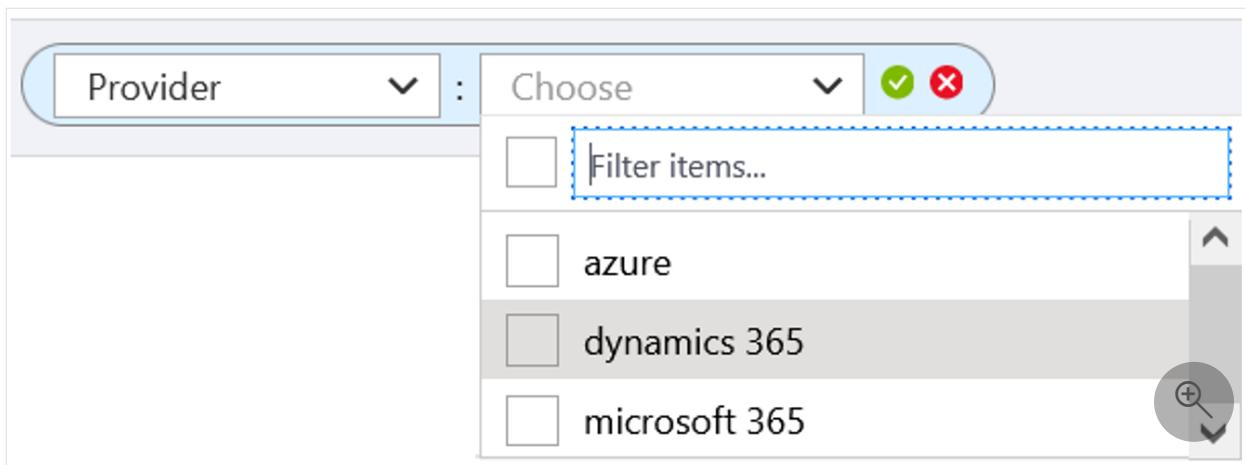
View your Azure and AWS costs together

To view Azure and AWS costs together, you use management group scopes in Azure.

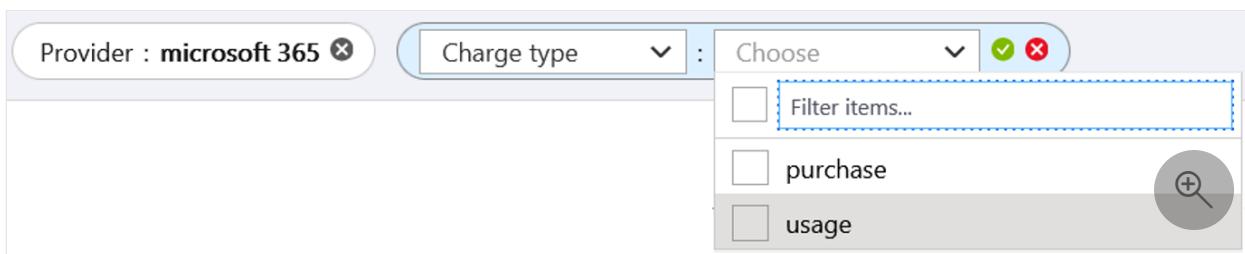
1. Create a management group or select an existing one.
2. Assign the existing Azure subscriptions that you need to the management group.
3. Assign the *same* management group to the linked account of the connector.
4. Go to cost analysis and select **Accumulated costs**.
5. Select **Group by - Provider**.

View New Commerce license and consumption costs

You can view your New Commerce license and consumption products along with your Azure charges in Cost analysis. Select the filter list, then select **Provider**, and then choose from the list of options. For example, Microsoft 365 and Dynamics 365.



You can narrow to specific seat-based or consumption charges by using the **Charge type** filter and selecting values of **Purchase** or **Usage**.



Currently, purchasing New Commerce products is only available for Partners.

View cost breakdown by Azure service

Viewing costs by an Azure service can help you to better understand the parts of your infrastructure that cost the most. For example, VM compute costs might be small. Yet you might accrue significant networking costs because of the amount of information emitting from the VMs. Understanding the primary cost drivers of your Azure services is essential so that you can adjust service usage, as needed.

1. In the Azure portal, navigate to cost analysis for your scope. For example: **Cost Management + Billing > Cost Management > Cost analysis**.
2. Select **Cost by service** and then group by **Service tier**.
3. Change the view to **Table**.

The screenshot shows the Azure Cost Management + Billing portal. The left sidebar is expanded to show 'Cost Management + Billing' and 'Cost analysis' is selected. The main area displays the 'Cost by service' view. Key metrics shown are ACTUAL COST (USD) of \$44,423.98 and FORECAST: CHART VIEW ON Jul-Sep 2019 of \$13,000/mo. The table below shows costs grouped by Service tier, with a magnifying glass icon over the last row.

Publisher type	Charge type	Service name	Service tier	Cost
azure	usage	log analytics	all	\$11,053.43
azure	usage	virtual machines	dv2/dsv2 series	\$7,509.44
azure	usage	storage	premium ssd managed disks	\$4,302.19
azure	usage	virtual machines	dv2/dsv2 series windows	\$2,698.49
azure	usage	storage	premium page blob	\$2,570.43
azure	usage	azure firewall	all	\$1,932.05
azure	usage	azure app service	standard plan	\$1,545.60
azure	usage	azure cosmos db	all	\$1,410.79
azure	usage	virtual machines	dv3/dsv3 series windows	\$1,330.15
azure	usage	virtual machines	a series windows	\$816.66
azure	usage	vnet gateway	high performance gateway	\$757.17
azure	usage	storage	standard hdd managed disks	\$735.51

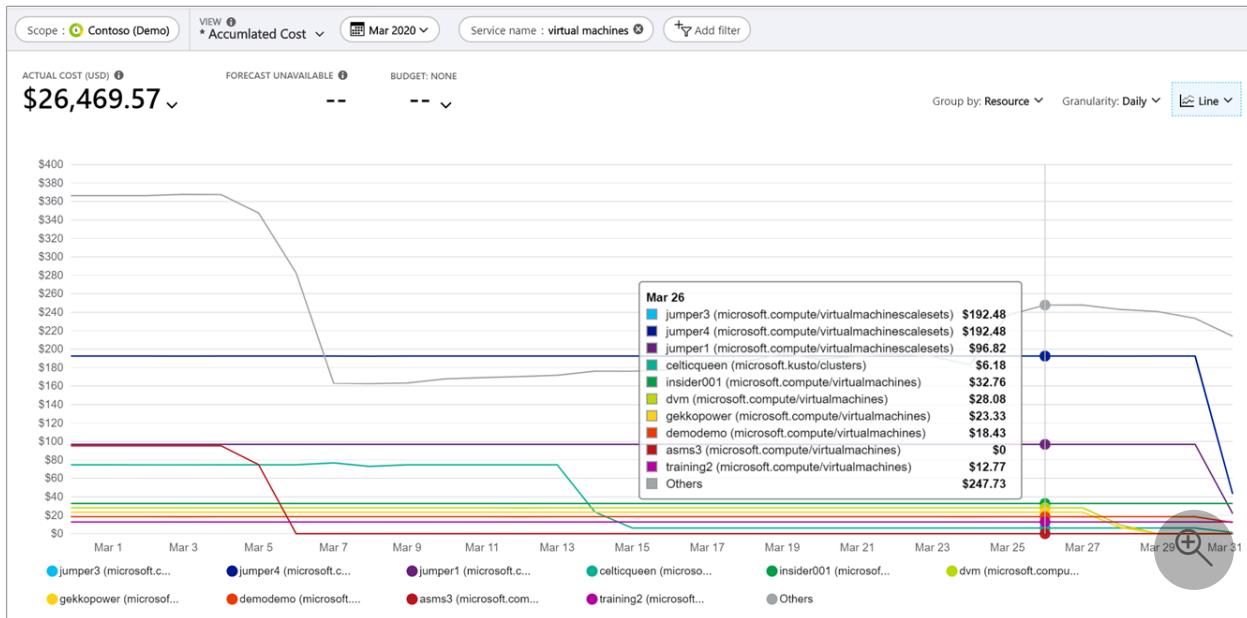
Review invoiced charges in Cost analysis

To view your invoice details in the Azure portal, navigate to Cost analysis for the scope associated with the invoice that you're analyzing. Select the **Invoice details** view. Invoice details show you the charges as seen on the invoice.

The screenshot shows the Azure Cost Management + Billing portal. The left sidebar is expanded to show 'Cost Management + Billing' and 'Cost analysis' is selected. The main area displays the 'Invoice details' view. Key metrics shown are ACTUAL COST (USD) of \$91,412.07 and FORECAST UNAVAILABLE of \$100/mo. The table below shows charges grouped by Meter, with a magnifying glass icon over the last row.

Publisher type	Charge type	Service name	Service tier	Meter	Part Number	Cost
azure	usage	virtual machines	fsv2 series windows vm	f72s v2	aad-10036	\$11,634.61
azure	usage	azure stack hub	azure stack-windows vm	1 core	aaa-56070	\$9,432.73
azure	usage	hdinsight	hdinsight d series	d4	n7h-01995	\$7,455.47
azure	usage	expressroute	expressroute	premium metered data 10 gbp...	j2q-00792	\$6,398.39
azure	usage	virtual machines licenses	sql server ent	16 vcpu vm license	n7h-07116	\$3,936.18
azure	usage	virtual machines	dv2 series windows vm	d15 v2/ds15 v2	aaa-14527	\$2,926.18
azure	usage	sql database	sql db single/elastic pool gen ...	vcore	aad-17849	\$2,832.64

Viewing invoice details, you can identify the service that has unexpected costs and determine which resources are directly associated with the resource in Cost analysis. For example, if you want to analyze charges for the Virtual Machines service, navigate to the **Accumulated cost** view. Then, set the granularity to **Daily** and filter charges **Service name: Virtual machines** and group charges by **Resource**.



View cost breakdown by Azure resource

Your services are built with Azure resources. Reviewing costs based on resources can help you quickly identify your primary cost contributors. If a service has resources that are too expensive, consider making changes to reduce your costs.

The view is only available for subscription and resource group scopes.

1. In the Azure portal, navigate to cost analysis for your scope. For example: **Cost Management + Billing > Cost Management > Cost analysis**.
2. Select **Cost by resource**.
3. Change the view to **Table**.

The screenshot shows the Azure Cost Analysis blade with the 'Cost by resource' view selected. It displays the actual cost as '\$5,638.76' and the budget as '\$13,000 /mo'.

The table lists resources along with their details and costs:

Resource	Resource Type	Region	Resource Group	Tags	Cost
hybrid-k8s-azmon	Log Analytics workspace	us east	hybrid-k8s-azmon-rg	businessowner:, compos...	\$1,704.04
contosoretail-it	Log Analytics workspace	us east	contosoazurehq	businessowner:, compos...	\$615.93
yqidlucontosoeusasrcache	Storage account	us south central	contosorecoveryvaultseus	environment:prod	\$311.29
defaultworkspace-e4272...	Log Analytics workspace	us east	defaultresourcegroup-eus	businessowner:, compos...	\$295.65
contosojumpboxazfirewall	Firewall	us south central	contosojumpboxfirewall	Not applicable	\$160.15
chqdisksk7owxak6txjx7u	Storage account	us south central	contosoazurehq	displayname:vmstorage	\$107.51
vmscaleset	Virtual machine scale set	us east	contosochefautomatevms	x-application:rhel-harde...	\$78.71

View cost breakdown by selected dimensions

Dimensions allow you to organize your costs based on various metadata values shown in your charges. For example, you could group your costs by location.

1. In the Azure portal, navigate to cost analysis for your scope. For example: **Cost Management + Billing > Cost Management > Cost analysis**.
2. Select the **Group by** filter.

ACTUAL COST (USD) **\$5,638.76**

FORECAST: CHART VIEW ON **--**

BUDGET: DEVTESTSPENDLIMIT **\$13,000 /mo**

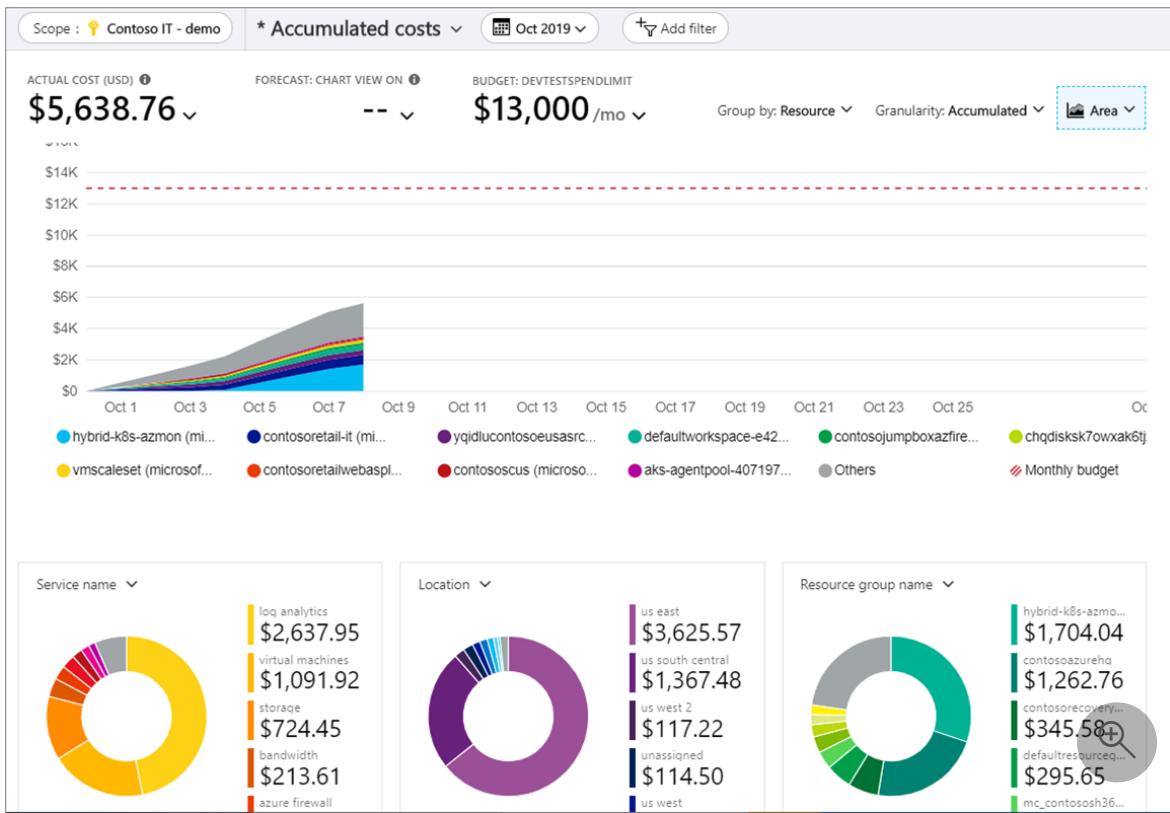
Group by: Resource

Resource	Resource type	Location
hybrid-k8s-azmon	Log Analytics workspace	us east
contosoretail-it	Log Analytics workspace	us east
yqidlucontosoeusasrcac...	Storage account	us south central
defaultworkspace-e427...	Log Analytics workspace	us east
contosojumpboxazfire...	Firewall	us south central
chqdisksk7owxak6tjx7u	Storage account	us south central

Filter items... 469 rows

None
Billing period
Charge type
Frequency
InvoiceNumber
Location
Meter
Meter category
Meter subcategory

3. Optionally, you save the view for later use.
4. Select a pie chart below the graph to view more detailed data.

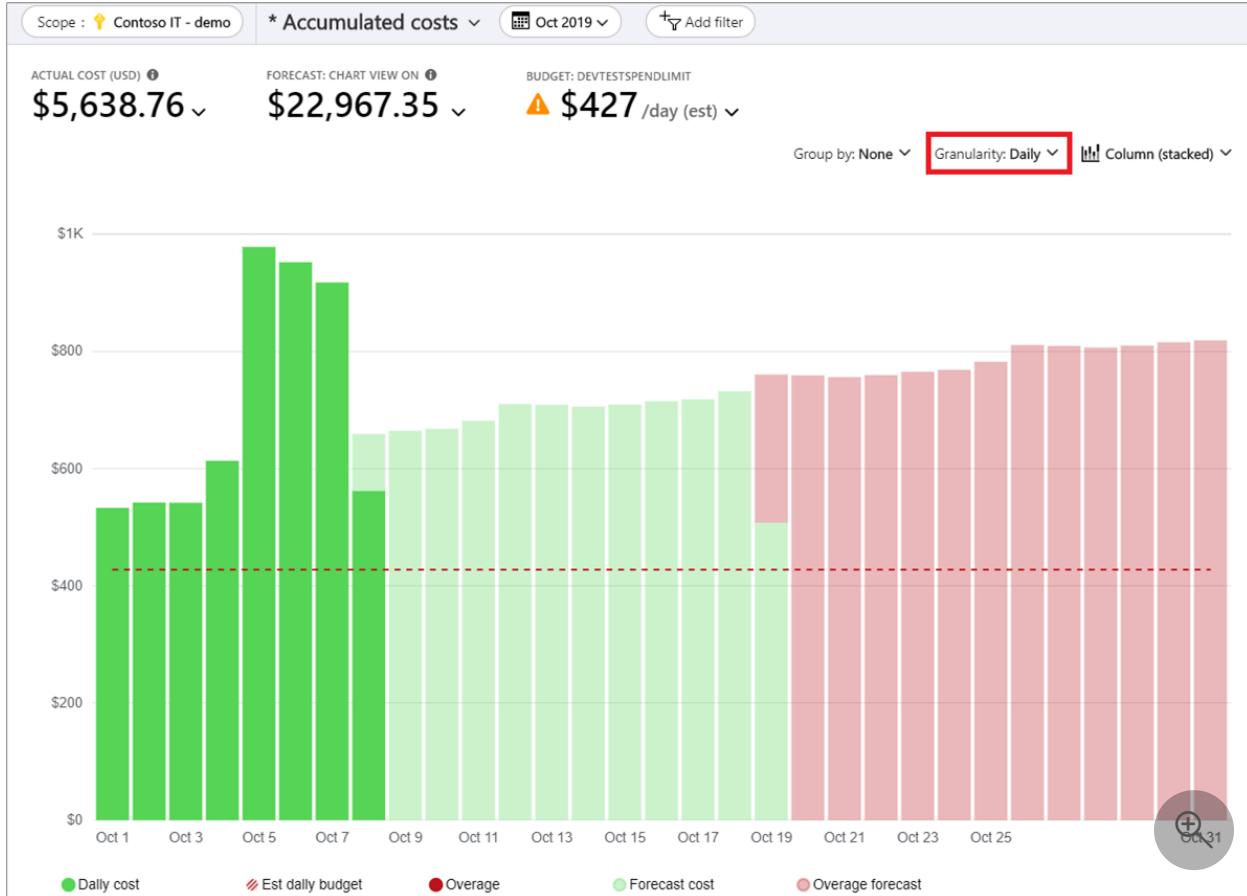


View costs per day or by month

Looking at daily and monthly costs can help you to better understand if there's a time of the week or year where your costs are higher. If you have more customer traffic in a

holiday period, does that lead to a corresponding increase in your Azure costs? Is Friday a more costly day than Monday?

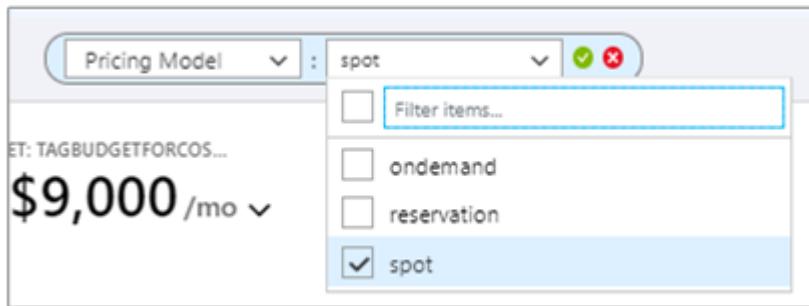
1. In the Azure portal, navigate to cost analysis for your scope. For example: **Cost Management + Billing > Cost Management > Cost analysis**.
2. Set the **Granularity** to **Monthly** or **Daily**.



View your Spot VM charges

Spot VMs can provide large cost savings for workloads that can handle interruptions. Workloads are run on unused Azure capacity. Since they can be evicted at any time, Spot VMs get a significant discount. Use the following steps to view your Spot VM charges.

1. In the Azure portal, navigate to cost analysis for your scope. For example, **Cost Management + Billing > Cost Management > Cost analysis**.
2. Add a filter for **Pricing Model: Spot**.



The Pricing Model dimension is also used to view on demand and reservation charges.

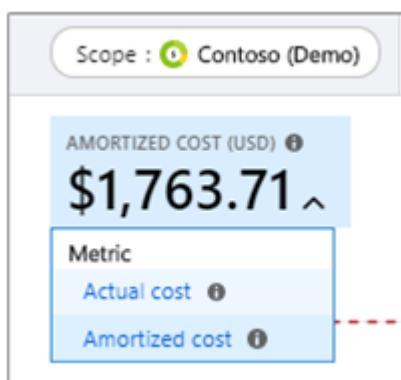
View your reservation charges

Reserved instances provide a way for you to save money with Azure. With reservations, you spend money up front for a given number of resources over time. Cost analysis shows the charges as they appear on your bill. The charges are shown as actual costs or amortized over the course of your reservation period.

ⓘ Note

Although you can buy a reservation with a pay-as-you-go (MS-AZR-0003P) subscription, Cost Analysis doesn't support viewing amortized reservation costs. If you try to view costs with the **Amortized cost** metric, you'll see the same results as **Actual Cost**.

1. In the Azure portal, navigate to cost analysis for your scope. For example, **Cost Management + Billing > Cost Management > Cost analysis**.
2. Add a filter for **Pricing Model: Reservation**.
3. Under **Scope** and next to the cost shown, select the down arrow symbol, select either **Actual cost** or **Amortized cost** metric.



Each metric affects how data is shown for your reservation charges.

Actual cost - Shows the purchase as it appears on your bill. For example, if you bought a one-year reservation for \$1200 in January, cost analysis shows a \$1200 cost in the

month of January for the reservation. It doesn't show a reservation cost for other months of the year. If you group your actual costs by VM, then a VM that received the reservation benefit for a given month would have zero cost for the month.

Amortized cost - Shows a reservation purchase split as an amortized cost over the duration of the reservation term. Using the same example above, cost analysis shows a varying cost for each month throughout the year, because of the varying number of days in a month. If you group costs by VM in this example, you'd see cost attributed to each VM that received the reservation benefit.

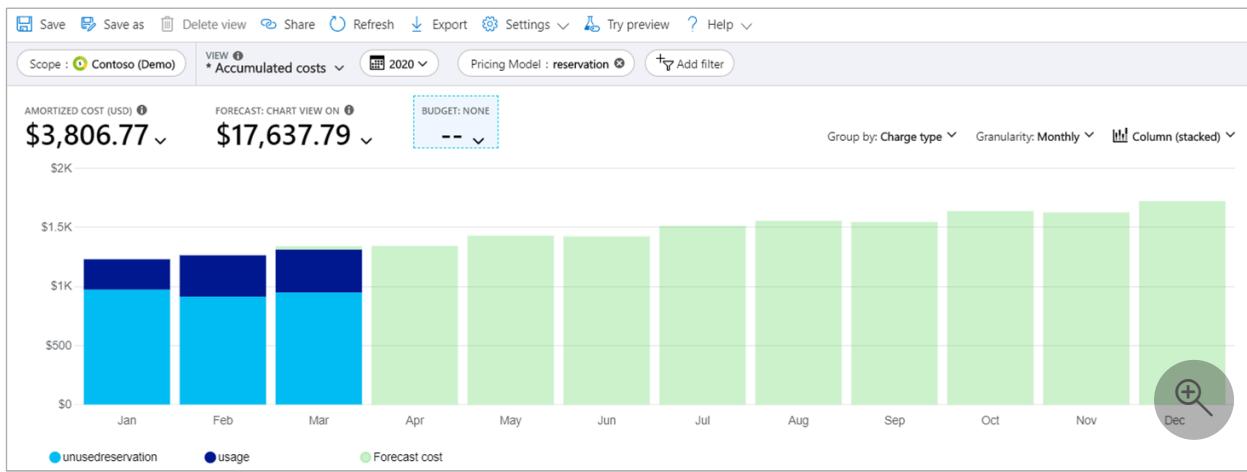
View your reservation utilization

After you buy a reservation, it's important to track its utilization so that you get what you paid for. For example, if you bought 10 VMs for a year and only use five of them, then essentially half of the purchase is wasted. There are two different ways to assess your utilization:

View unused RI costs in cost analysis

To identify how much cost is currently being wasted each month for your reservation purchase, follow the steps below.

1. In the Azure portal, navigate to cost analysis for the scope where your reservation is applied. For example, **Cost Management + Billing > Cost Management > Cost analysis**.
2. Add a filter for **Pricing Model: Reservation**.
3. Select the **Amortized Cost** view.
4. Set the granularity to **Monthly**.
5. Set the time period to the current year or your reservation term.
6. Set the chart type to **Column (stacked)**.
7. Group charges by **Charge Type**.
8. Review the results for **unusedreservation** values.



View utilization in Reservations

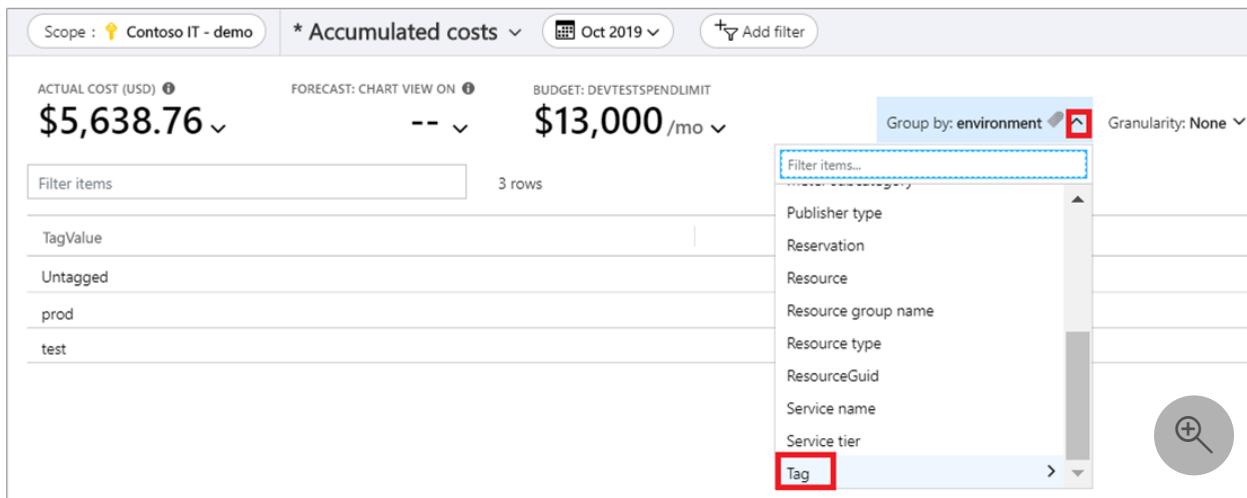
For detailed instructions, see [Optimize reservation use](#).

View costs for a specific tag

Many Azure users apply tags to their resources such as a cost center or development environment (production and test) to better categorize charges. Tags appear as a dimension in cost analysis. You can use the dimension to gain insights into your custom tagging categorizations.

Support for tags applies to usage reported *after* the tag was applied to the resource. Tags aren't applied retroactively for cost rollups.

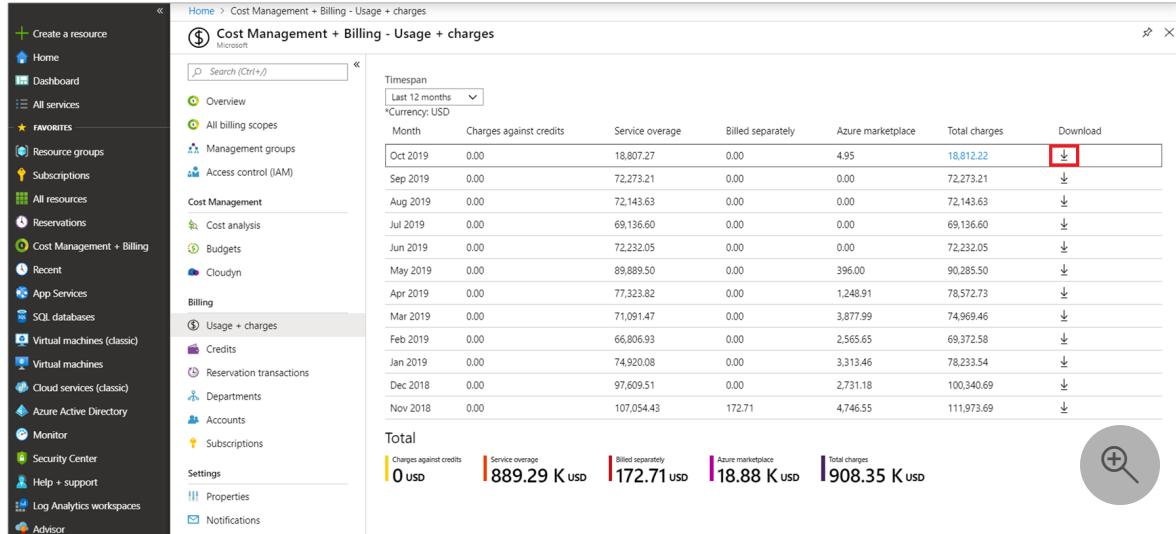
1. In the Azure portal, navigate to cost analysis for your scope. For example: **Cost Management + Billing > Cost Management > Cost analysis**.
2. Select **Group by** for your tag.



Download your usage details

Your usage details report file, in CSV format, provides a breakdown of all the charges that accrued towards an invoice. You can use the report to compare it to, and better understand, your invoice. Each billed charge on your invoice corresponds to broken-down charges in the usage report.

1. In the Azure portal, navigate to the **Usage and Charges** tab for a billing account or subscription. For example: **Cost Management + Billing > Billing > Usage + charges**.
2. Select the line item to download from and then select the download symbol.



The screenshot shows the Azure portal's navigation bar on the left, with the 'Cost Management + Billing' section selected. On the right, the 'Usage + charges' tab is active. A table displays monthly charges against credits, service overage, and total charges. The 'Download' column contains a small downward arrow icon, which is highlighted with a red box. The table includes columns for Month, Charges against credits, Service overage, Billed separately, Azure marketplace, Total charges, and Download.

Month	Charges against credits	Service overage	Billed separately	Azure marketplace	Total charges	Download
Oct 2019	0.00	18,807.27	0.00	4.95	18,812.22	
Sep 2019	0.00	72,273.21	0.00	0.00	72,273.21	
Aug 2019	0.00	72,143.63	0.00	0.00	72,143.63	
Jul 2019	0.00	69,136.60	0.00	0.00	69,136.60	
Jun 2019	0.00	72,232.05	0.00	0.00	72,232.05	
May 2019	0.00	89,889.50	0.00	396.00	90,285.50	
Apr 2019	0.00	77,323.82	0.00	1,248.91	78,572.73	
Mar 2019	0.00	71,091.47	0.00	3,877.99	74,969.46	
Feb 2019	0.00	66,806.93	0.00	2,565.65	69,372.58	
Jan 2019	0.00	74,920.08	0.00	3,313.46	78,233.54	
Dec 2018	0.00	97,609.51	0.00	2,731.18	100,340.69	
Nov 2018	0.00	107,054.43	172.71	4,746.55	111,973.69	

Total

Charges against credits 0 K usd	Service overage 889.29 K usd	Billed separately 172.71 K usd	Azure marketplace 18.88 K usd	Total charges 908.35 K usd
------------------------------------	---------------------------------	-----------------------------------	----------------------------------	-------------------------------



3. Select the usage file to download.

The screenshot shows a modal window titled "Download Usage + Charges" from October 2019. It displays two sections: "Usage Details" and "Usage Details Version 2".

Usage Details: Describes the Usage Detail csv as offering a daily breakdown of consumed quantities and estimated charges by an Enrollment. A blue "Download csv" button is present.

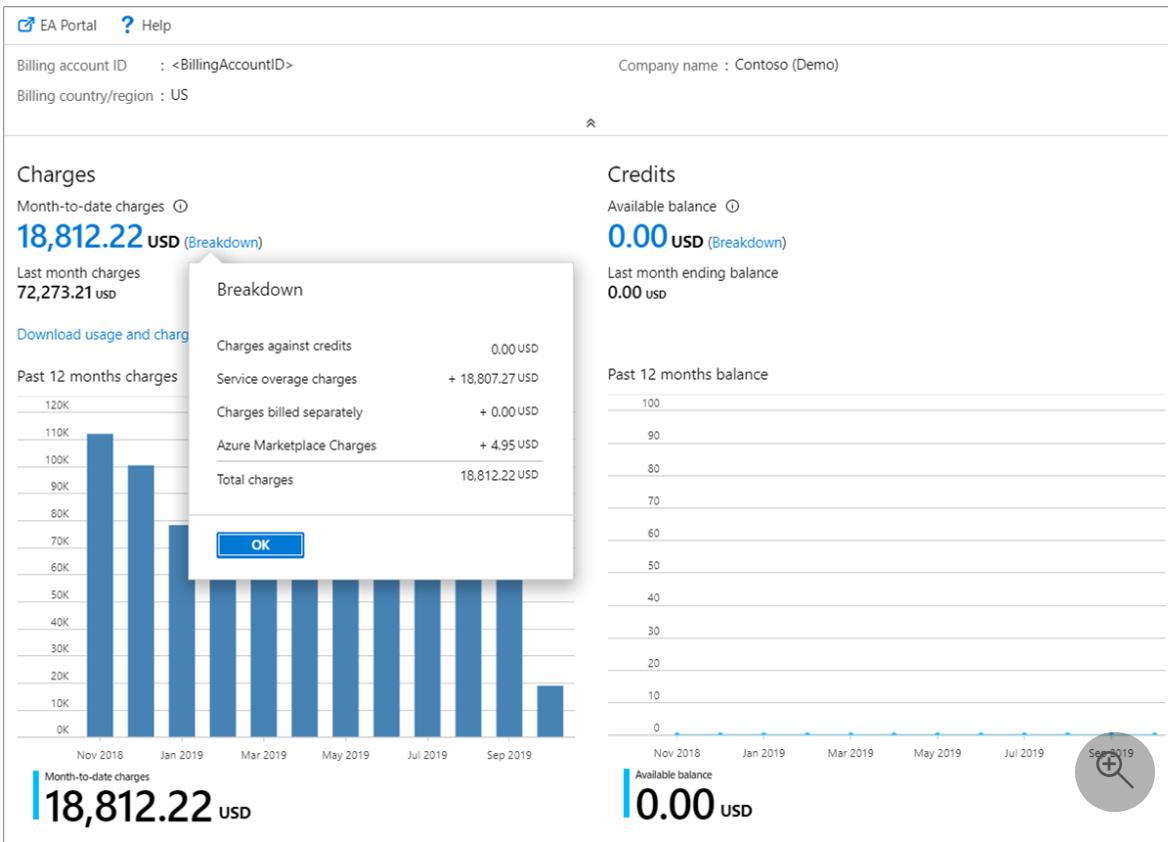
Usage Details Version 2: Describes the file as providing richer usage and purchase data, including all usage, RI purchases, and RI amortization. It includes a dropdown menu set to "All Charges (usage and purchases)" and a blue "Download csv" button.

View monthly EA cost breakdown

Your EA enrollment accrues costs for your entire organization. Understanding how costs accrue and are invoiced over time helps you to engage the appropriate stakeholders to ensure that costs are managed responsibly.

Costs are only shown for your active enrollment. If you transferred an enrollment (inactive) to a new one (active), costs for the previous enrollment aren't shown in Cost Management.

1. In the Azure portal, navigate to **Cost Management + Billing > Overview**.
2. Select **Breakdown** for the current month and view your Azure Prepayment (previously called monetary commitment) burn down.



- Select the **Usage and Charges** tab and view the prior month's breakdown in the chosen timespan.

The screenshot shows the 'Cost Management + Billing - Usage + charges' page. The left sidebar includes links for Overview, All billing scopes, Management groups, Access control (IAM), Cost Management (Cost analysis, Budgets, Cloudyn), Billing (Usage + charges, Credits, Reservation transactions, Departments, Accounts, Subscriptions), Settings, and Properties. The main area shows a table of monthly charges from Oct 2018 to Nov 2018, with a total for each month and a cumulative total at the bottom. A search bar and a 'Timespan' dropdown are also present.

Month	Charges against credits	Service overage	Billed separately	Azure marketplace	Total charges	Download			
Oct 2019	0.00	18,807.27	0.00	4.95	18,812.22				
Sep 2019	0.00	72,273.21	0.00	0.00	72,273.21				
Aug 2019	0.00	72,143.63	0.00	0.00	72,143.63				
Jul 2019	0.00	69,136.60	0.00	0.00	69,136.60				
Jun 2019	0.00	72,232.05	0.00	0.00	72,232.05				
May 2019	0.00	89,889.50	0.00	396.00	90,285.50				
Apr 2019	0.00	77,323.82	0.00	1,248.91	78,572.73				
Mar 2019	0.00	71,091.47	0.00	3,877.99	74,969.46				
Feb 2019	0.00	66,806.93	0.00	2,565.65	69,372.58				
Jan 2019	0.00	74,920.08	0.00	3,313.46	78,233.54				
Dec 2018	0.00	97,609.51	0.00	2,731.18	100,340.69				
Nov 2018	0.00	107,054.43	172.71	4,746.55	111,973.69				
Total									
Charges against credits	0 usd	Service overage	889.29 K usd	Billed separately	172.71 usd	Azure marketplace	18.88 K usd	Total charges	908.35 K usd

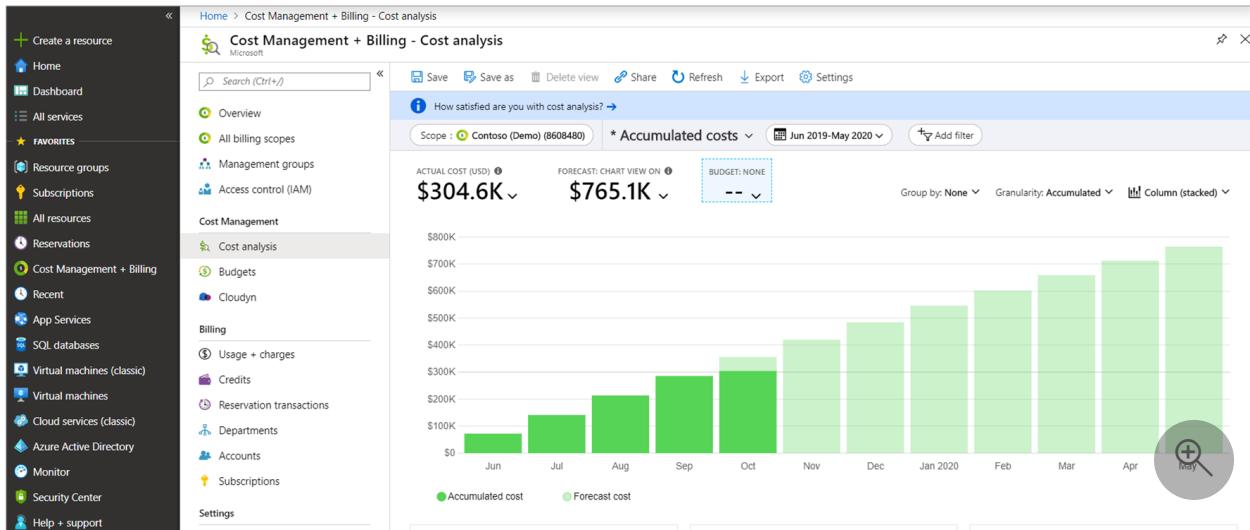
View enrollment monthly cost by term

Use a graphical view of your enrollment's monthly costs to understand the cost trends and invoiced amounts for a given period.

- In the Azure portal, navigate to cost analysis for your scope. For example: **Cost Management + Billing > Cost Management > Cost analysis**.
- Select your enrollment and set the enrollment term.

3. Set the granularity to monthly and then set the view to Column (stacked).

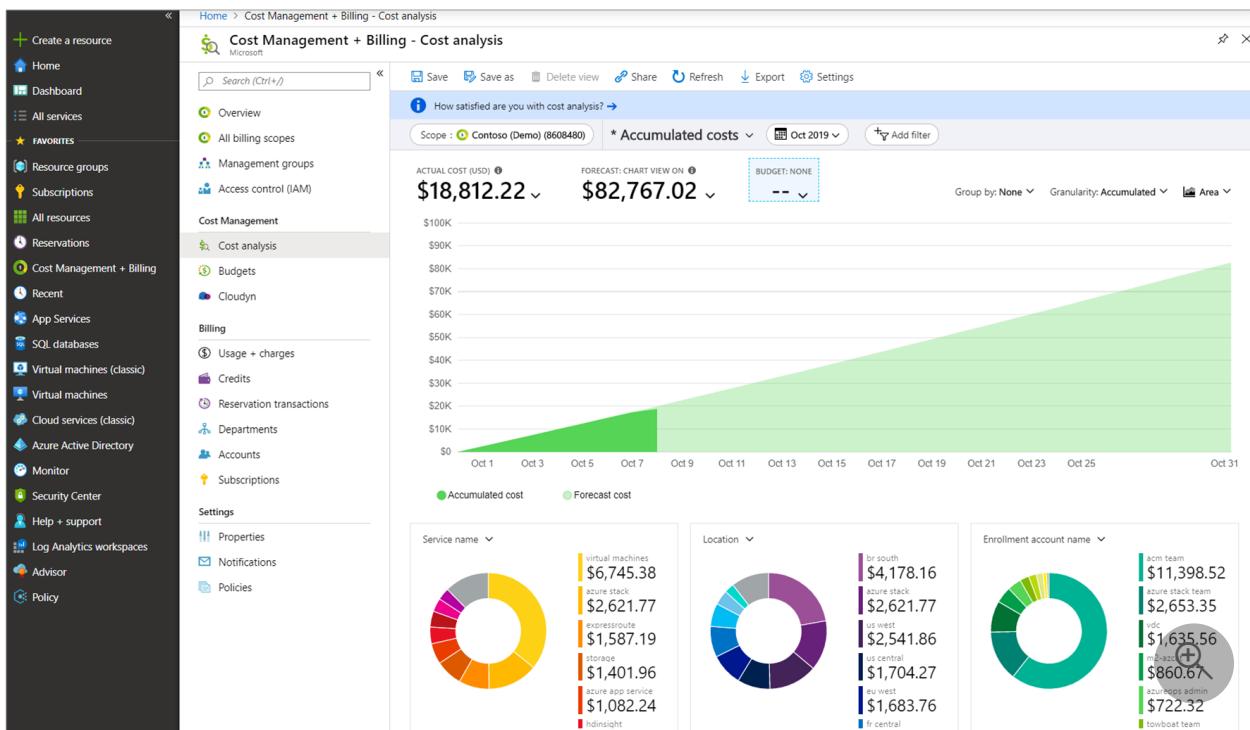
You can group by and filter your data for a more detailed analysis.



View EA enrollment accumulated costs

View the net accumulated charges over time to understand overall expenditures for your organization for a given period.

1. In the Azure portal, navigate to cost analysis for your scope. For example: **Cost Management + Billing > Cost Management > Cost analysis**.
2. Select your enrollment and then view your current accumulated costs.



Next steps

- If you haven't already completed the first quickstart for Cost Management, read it at [Start analyzing costs](#).
- Read the [Cost Management documentation](#).

Create visuals and reports with the Azure Cost Management connector in Power BI Desktop

Article • 03/20/2023

You can use the Azure Cost Management connector for Power BI Desktop to make powerful, customized visualizations and reports that help you better understand your Azure spend. The Azure Cost Management connector currently supports customers with a direct [Microsoft Customer Agreement](#) or an [Enterprise Agreement \(EA\)](#).

The Azure Cost Management connector doesn't support pay-as-you-go Microsoft Customer Agreements or indirect Microsoft Customer Agreements. Microsoft Partner Agreements are also not supported. If you have an unsupported agreement, you can use Exports to save the cost data to a share and then connect to it using Power BI. For more information, see [Tutorial - Create and manage exported data from Azure Cost Management](#).

The Azure Cost Management connector uses OAuth 2.0 for authentication with Azure and identifies users who are going to use the connector. Tokens generated in this process are valid for a specific period. Power BI preserves the token for the next login. OAuth 2.0, is a standard for the process that goes on behind the scenes to ensure the secure handling of these permissions. To connect, you must use an [Enterprise Administrator](#) account for Enterprise Agreements, or have [appropriate permissions](#) at the billing account or billing profile levels for Microsoft Customer Agreements.

Note

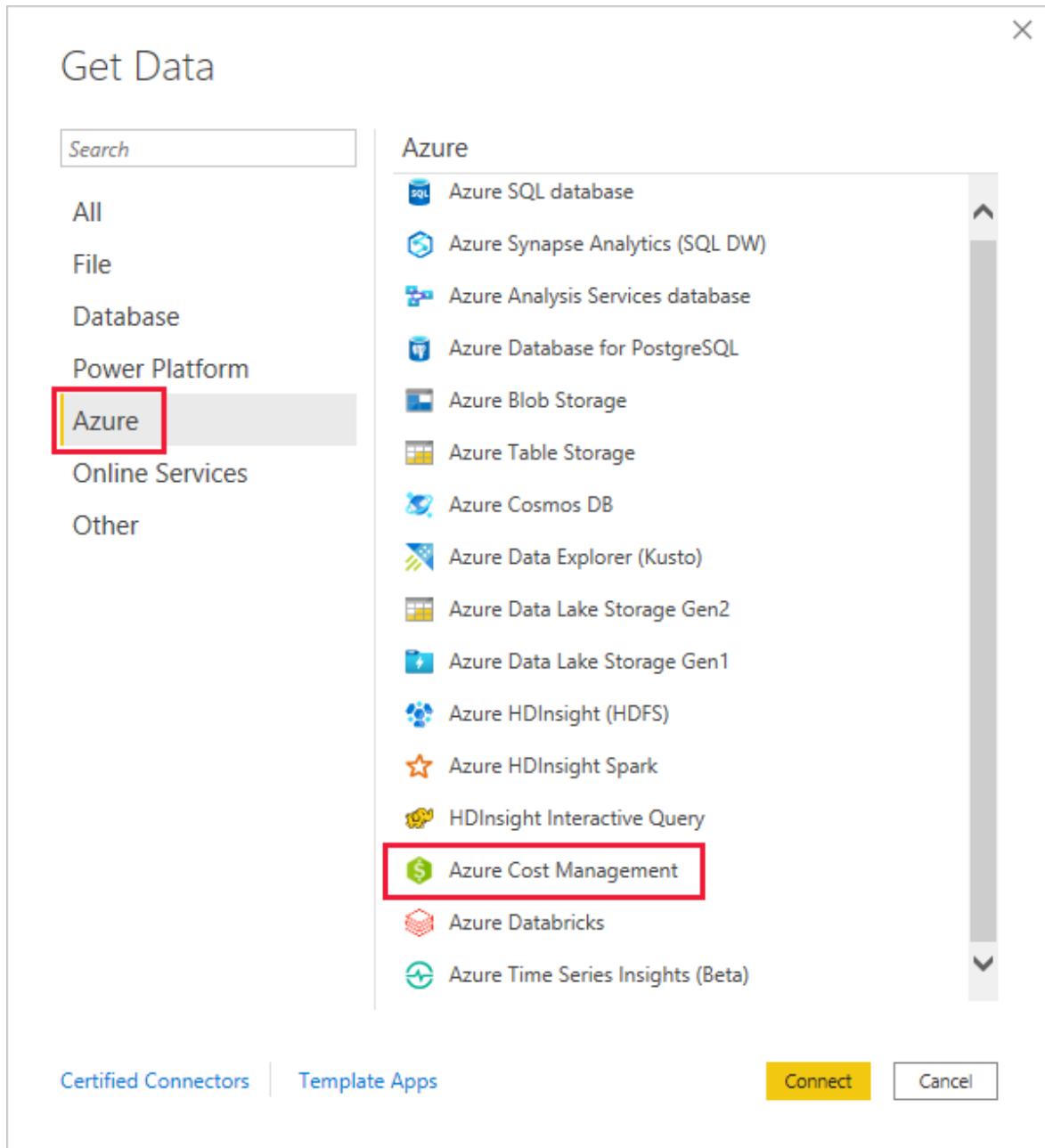
This connector replaces the previously available [Azure Consumption Insights](#) and [Azure Cost Management \(Beta\)](#) connectors. Any reports created with the previous connector must be recreated using this connector.

Connect using Azure Cost Management

To use the Azure Cost Management connector in Power BI Desktop, take the following steps:

1. In the Home ribbon, select **Get Data**.
2. Select **Azure** from the list of data categories.

3. Select Azure Cost Management.



4. In the dialog that appears, for the **Choose Scope** drop down, use **Manually Input Scope for Microsoft Customer Agreements**, or use **Enrollment Number** for Enterprise Agreements (EA).

Connect to a Microsoft Customer Agreement account

This section describes the steps necessary to connect to a Microsoft Customer Agreement account.

Connect to a billing account

To connect to a billing account, you need to retrieve your **Billing account ID** from the Azure portal:

1. In the [Azure portal](#), navigate to **Cost Management + Billing**.
2. Select your Billing profile.
3. Under **Settings** in the menu, select **Properties** in the sidebar.
4. Under **Billing profile**, copy the ID.

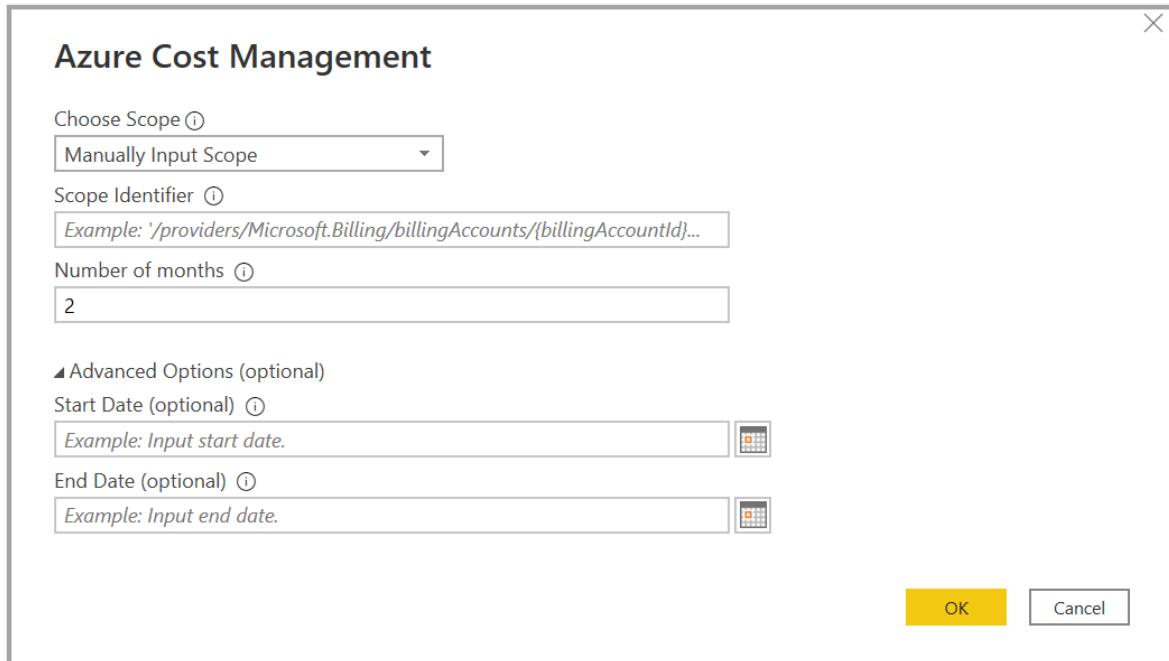
The screenshot shows the Azure portal interface for managing a billing account. The top navigation bar includes 'Home', 'Cost Management + Billing', and 'Test billing acct'. The main title is 'Test billing acct | Properties' under the 'Billing account' section. The left sidebar lists several sections: 'Overview', 'Access control (IAM)', 'Billing scopes', 'Cost management' (with sub-options for 'Cost analysis', 'Cost alerts', 'Budgets', and 'Advisor recommendations'), 'Billing' (with sub-options for 'Invoices', 'Payment methods', 'Reservation transactions', and 'Billing profiles'), 'Products + services' (with sub-options for 'Azure subscriptions' and 'Recurring charges'), 'Settings' (with sub-options for 'Properties' - which is highlighted with a grey background, 'Exports', and 'Cost allocation (preview)'), and 'Support + troubleshooting' (with sub-option for 'New support request'). The 'General' tab is selected in the top navigation bar. On the right side, there are details for the 'Billing profile': 'ID' is 'TestUser', 'Name' is 'TestUser', 'Type' is 'Microsoft Customer Agreement', and 'Sold-to' is 'Microsoft Redmond wa 98052 US'. There is also a 'Billing account owner' listed as 'TestUser'. A red box highlights the 'Copy' icon next to the 'ID' field. Below the 'General' tab, there is a link 'Update name' under 'Name' and another link 'Update sold-to' under 'Sold-to'.

5. For **Choose Scope**, select **Manually Input Scope** and input the connection string as shown in the example below, replacing `{billingAccountId}` with the data copied from the previous steps.

```
/providers/Microsoft.Billing/billingAccounts/{billingAccountId}
```

Alternatively, for **Choose Scope**, select **Enrollment Number** and input the Billing Account ID string as copied from the previous steps.

6. Enter the number of months and select **OK**.



The screenshot shows the 'Demo Billing Profile | Properties' page in the Azure portal. On the left, there's a navigation menu with sections like Overview, Access control (IAM), Billing scopes, Cost management, Cost analysis, Cost alerts, Budgets, Advisor recommendations, Billing, Invoices, Payment methods, Payment history, Reservation transactions, Invoice sections, Products + services, Azure subscriptions, Recurring charges, Settings, and Properties. The 'Properties' item is currently selected. The main content area is divided into two main sections: 'Billing profile' and 'Billing account'. The 'Billing profile' section contains fields for Name (Demo Billing Profile), PO number (Not available), Billing address (Redmond, WA, 98052, US), Enabled Azure plans (Microsoft Azure Plan), My role (Billing profile reader), Tags (No tags added), and a link to Add or update tags. The 'Billing account' section contains fields for ID (highlighted with a red box) and Type (Microsoft Customer Agreement). There are also '...' and edit icons for both sections.

6. For **Choose Scope**, select **Manually Input Scope** and input the connection string as shown in the example below, replacing `{billingAccountId}` and `{billingProfileId}` with the data copied from the previous steps.

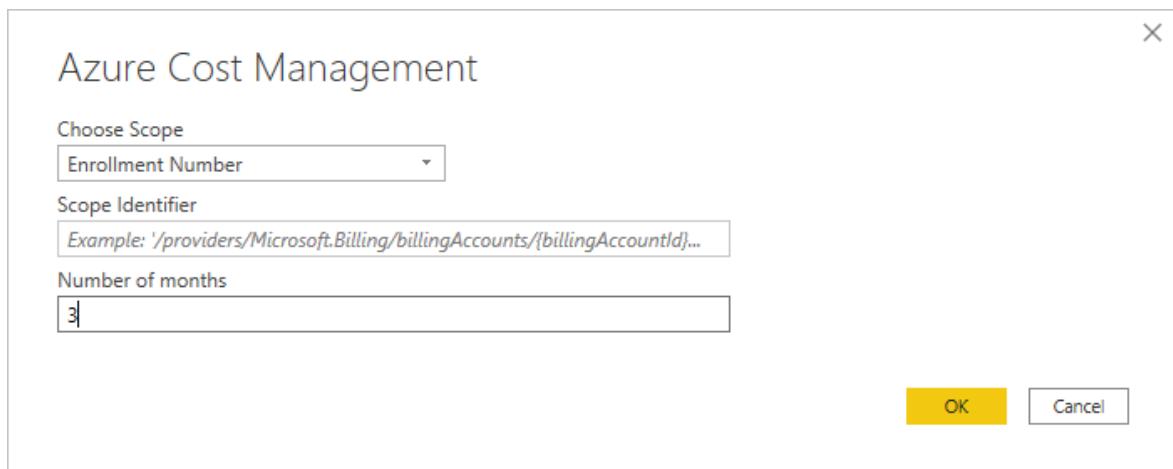
```
/providers/Microsoft.Billing/billingAccounts/{billingAccountId}/billingProfile  
s/{billingProfileId}
```

7. Enter the number of months and select **OK**.
8. When prompted, sign in with your Azure user account and password. You must have access to the Billing profile to successfully access the billing profile data.

Connect to an Enterprise Agreement account

To connect with an Enterprise Agreement (EA) account, you can get your enrollment ID from the Azure portal:

1. In the [Azure portal](#), navigate to **Cost Management + Billing**.
2. Select your billing account.
3. On the **Overview** menu, copy the **Billing account ID**.
4. For **Choose Scope**, select **Enrollment Number** and paste the billing account ID from the previous step.
5. Enter the number of months and then select **OK**.



6. When prompted, sign in with your Azure user account and password. You must use an Enterprise Administrator account for Enterprise Agreements.

Data available through the connector

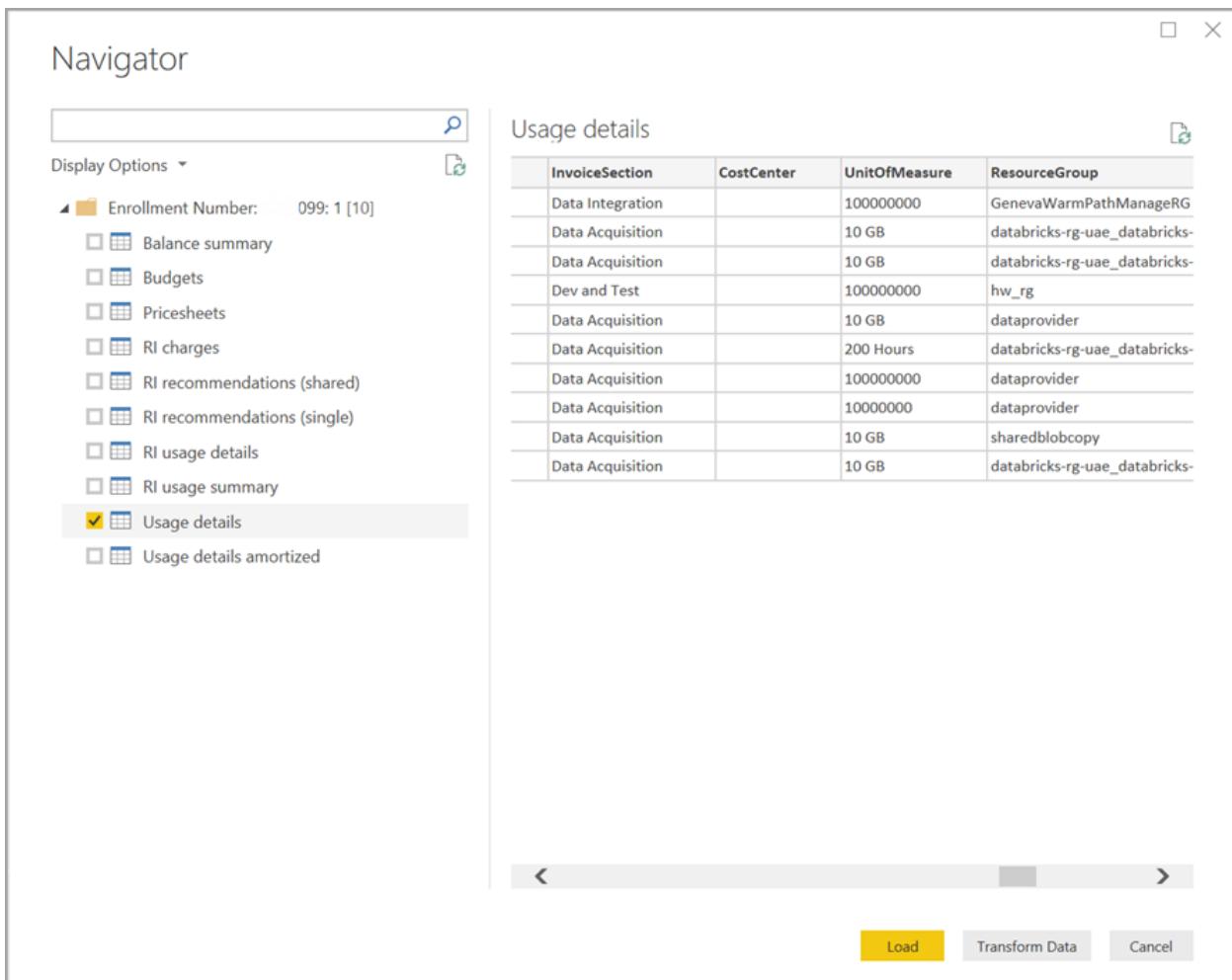
Once you successfully authenticate, a **Navigator** window appears with the following available data tables:

Table	Account Type	Supported Scopes	Description
Balance summary	EA only	EA Enrollment	Summary of the balance for the current billing month for Enterprise Agreements (EA).
Billing events	MCA only	Billing Profile	Event log of new invoices, credit purchases, etc. Microsoft Customer Agreement only.

Table	Account Type	Supported Scopes	Description
Budgets	EA, MCA	EA Enrollment,MCA Billing Account,MCA Billing Profile	Budget details to view actual costs or usage against existing budget targets.
Charges	MCA only	MCA Billing Profile	A month-level summary of Azure usage, Marketplace charges, and charges billed separately. Microsoft Customer Agreement only.
Credit lots	MCA only	MCA Billing Profile	Azure credit lot purchase details for the provided billing profile. Microsoft Customer Agreement only.
Pricesheets	EA, MCA	EA Enrollment,MCA Billing Profile	Applicable meter rates for the provided billing profile or EA enrollment.
RI charges	EA, MCA	EA Enrollment,MCA Billing Profile	Charges associated to your Reserved Instances over the last 24 months. This table is in the process of being deprecated, please use RI transactions
RI recommendations (shared)	EA, MCA	EA Enrollment,MCA Billing Profile	Reserved Instance purchase recommendations based on all your subscription usage trends for the last 30 days.
RI recommendations (single)	EA, MCA	EA Enrollment,MCA Billing Profile	Reserved Instance purchase recommendations based on your single subscription usage trends for the last 30 days.
RI transactions	EA, MCA	EA Enrollment,MCA Billing Profile	List of transactions for reserved instances on billing account scope.
RI usage details	EA, MCA	EA Enrollment,MCA Billing Profile	Consumption details for your existing Reserved Instances over the last month.
RI usage summary	EA, MCA	EA Enrollment,MCA Billing Profile	Daily Azure reservation usage percentage.

Table	Account Type	Supported Scopes	Description
Usage details	EA, MCA	EA Enrollment,MCA Billing Account,MCA Billing Profile	A breakdown of consumed quantities and estimated charges for the given billing profile on EA enrollment.
Usage details amortized	EA, MCA	EA Enrollment,MCA Billing Account,MCA Billing Profile	A breakdown of consumed quantities and estimated amortized charges for the given billing profile on EA enrollment.

You can select a table to see a preview dialog. You can select one or more tables by selecting the boxes beside their name and then select **Load**.



When you select **Load**, the data is loaded into Power BI Desktop.

When the data you selected is loaded, the data tables and fields are shown in the **Fields** pane.

Considerations and limitations

The following considerations and limitations apply to the Azure Cost Management data connector:

- Data row requests exceeding one million rows isn't supported by Power BI. Instead, you can try using the export feature described in [create and manage exported data in Azure Cost Management](#).
- The Azure Cost Management data connector doesn't work with Office 365 GCC customer accounts.
- **Data refresh:** The cost and usage data is typically updated and available in the Azure portal and supporting APIs within 8 to 24 hours, so we suggest you constrain Power BI scheduled refreshes to once or twice a day.
- **Data source reuse:** If you have multiple reports that are pulling the same data, and don't need additional report-specific data transformations, you should reuse the same data source, which would reduce the amount of time required to pull the Usage Details data.

For more information on reusing data sources, see the following:

- [Introduction to datasets across workspaces](#)
- [Create reports based on datasets from different workspaces](#)

You might receive a *400 bad request* from the **RI usage details** when you try to refresh the data if you've chosen date parameter greater than three months. To mitigate the error, take the following steps:

1. In Power BI Desktop, select **Home > Transform data**.
2. In Power Query Editor, select the **RI usage details** dataset and select **Advanced Editor**.
3. Update the Power Query code as shown in the following paragraph(s), which will split the calls into three-month chunks. Make sure you note and retain your enrollment number, or billing account/billing profile ID.

For EA use the following code update:

```
let
    enrollmentNumber = "<>",
    optionalParameters1 = [startBillingDataWindow = "-9",
    endBillingDataWindow = "-6"],
```

```

    source1 = AzureCostManagement.Tables("Enrollment Number",
enrollmentNumber, 5, optionalParameters1),
    riusagedetails1 = source1{[Key="riusagedetails"]}[Data],
    optionalParameters2 = [startBillingDataWindow = "-6",
endBillingDataWindow = "-3"],
    source2 = AzureCostManagement.Tables("Enrollment Number",
enrollmentNumber, 5, optionalParameters2),
    riusagedetails2 = source2{[Key="riusagedetails"]}[Data],
    riusagedetails = Table.Combine({riusagedetails1, riusagedetails2})
in
    riusagedetails

```

For Microsoft Customer Agreements use the following update:

```

let
    billingProfileId = "<>Billing Profile Id>>",
    optionalParameters1 = [startBillingDataWindow = "-9",
endBillingDataWindow = "-6"],
    source1 = AzureCostManagement.Tables("Billing Profile Id",
billingProfileId, 5, optionalParameters1),
    riusagedetails1 = source1{[Key="riusagedetails"]}[Data],
    optionalParameters2 = [startBillingDataWindow = "-6",
endBillingDataWindow = "-3"],
    source2 = AzureCostManagement.Tables("Billing Profile Id",
billingProfileId, 5, optionalParameters2),
    riusagedetails2 = source2{[Key="riusagedetails"]}[Data],
    riusagedetails = Table.Combine({riusagedetails1, riusagedetails2})
in
    riusagedetails

```

- Once you've updated the code with the appropriate update from the previous step, select **Done** and then select **Close & Apply**.

You might run into a situation where tags aren't working in the usage details or the tags column can't be transformed to json. This issue stems from the current UCDD api returning the tags column by trimming the start and end brackets, which results in Power BI being unable to transform the column because it returns it as a string. To mitigate this situation, take the following steps.

1. Navigate to **Query Editor**.
2. Select the *Usage Details* table.
3. In the right pane, the **Properties** pane shows the **Applied Steps**. You need to add a custom column to the steps, after the **Navigation** step.
4. From the menu, select **Add column > Add custom column**

5. Name the column, for example you could name the column *TagsInJson* or whatever you prefer, and then enter the following text in the query:

```
DAX  
  
``` = "{" & [Tags] & "}"
```

6. Completing the previous steps creates a new column of *tags* in the json format

7. You can now transfer and expand the column as you need to.

**Authentication issues encountered with Azure Active Directory guest accounts:** You may have the appropriate permissions to access the enrollment or billing account, but receive an authentication error similar to one of the following:

- *Access to the resource is forbidden*
- *We couldn't authenticate with the credentials provided. Please try again.*

These errors could be the result of having a user account in a different Azure Active Directory domain that has been added as a guest user.

For guest accounts: Use the following settings or options as you are prompted with the **authentication dialog** when connecting with the Cost Management Power BI connector:

1. Select **Sign-in**
2. Select the **Use another account** (bottom of the dialog)
3. Select **Sign-in options** (bottom of the dialog box)
4. Select **Sign into an organization**
5. For **Domain name**, provide the Fully Qualified Domain Name (FQDN) of the Azure Active Directory domain into which you've been added as a guest.
6. Then, for **Pick an account** select the user account that you've previously authenticated.

## Next steps

You can connect to many different data sources using Power BI Desktop. For more information, see the following articles:

- [What is Power BI Desktop?](#)
- [Data Sources in Power BI Desktop](#)
- [Shape and Combine Data with Power BI Desktop](#)
- [Connect to Excel workbooks in Power BI Desktop](#)
- [Enter data directly into Power BI Desktop](#)

# Analyze cost with the Cost Management Power BI App for Enterprise Agreements (EA)

Article • 04/06/2023

This article explains how to install and use the Cost Management Power BI app. The app helps you analyze and manage your Azure costs in Power BI. You can use the app to monitor costs, usage trends, and identify cost optimization options to reduce your expenditures.

The Cost Management Power BI app currently supports only customers with an [Enterprise Agreement](#).

The app limits customizability. If you want to modify and extend the default filters, views, and visualizations to customize for your needs, use [Cost Management connector in Power BI Desktop](#) instead. With the Cost Management connector you can join additional data from other sources to create customized reports to get holistic views of your overall business cost. The connector also supports Microsoft Customer Agreements.

## ⓘ Note

Power BI template apps don't support downloading the PBIX file.

## Prerequisites

- A [Power BI Pro license](#) is required to install and use the app.
- To connect to data, you must use an [Enterprise Administrator](#) account. The Enterprise Administrator (read only) role is supported.

## Installation steps

To install the app:

1. Open [Cost Management Power BI App](#).
2. On the Power BI AppSource page, select **Get it now**.
3. Select **Continue** to agree to the terms of use and privacy policy.

4. In the **Install this Power BI app** box, select **Install**.
5. If needed, create a workspace and select **Continue**.
6. When installation completes, notification appears saying that your new app is ready.
7. Select the app that you installed.
8. On the Getting started page, select **Connect your data**.

The screenshot shows a Microsoft Power BI web interface. The title bar says "AzureCostManagementApp - Pow BI". The address bar shows the URL "msit.powerbi.com/groups/me/apps/c0481d88-38d6-4004-90f2-a56d05f93530/reports/a4a10d05-6fb8-41...". The top navigation bar includes "Microsoft", "Power BI", and the workspace name "docs-test-2". On the right, it says "Azure Cost Management App | Data updated 2/19/21". Below the navigation is a toolbar with "File", "Export", "Share", "Chat in Teams", "Comment", "Subscribe", and more. A message "You're viewing this app with sample data." has a red box around the "Connect your data" link. The main content area is titled "Welcome to Azure Cost Management". It features a "Getting Started" section with a description: "The Azure Cost Management Power BI App is meant to assist in analyzing and managing your app to monitor costs, usage trends and identify cost optimization options to reduce your expe..." and a link to "Power BI Desktop". There is also a search icon.

9. In the dialog that appears, enter your EA enrollment number for **BillingProfileIdOrEnrollmentNumber**. Specify the number of months of data to get. Leave the default **Scope** value of **Enrollment Number**, then select **Next**.

**ⓘ Note**

The default value for Scope is **Enrollment Number**. Do not change the value, otherwise the initial data connection will fail.



## Connect to Azure Cost Management App

Get started setting up your app! Start by filling in the parameters. Then, you'll authenticate to all the data sources this app connects to.

### Parameters

Before connecting to your data, you must update the required parameters (\*).

#### BillingProfileIdOrEnrollmentNumber

Enter your EA account enrollment number

#### NumberOfMonths

Enter the number of months

#### Scope

[Go to the app documentation ↗](#)

**Next**

**Cancel**

10. The next installation step connects to your EA enrollment and requires an [Enterprise Administrator](#) account. Leave all the default values. Select **Sign in and**

connect.



## Connect to Azure Cost Management App

You are connecting to (1 of 2)

ExtensionDataSourceKind  
AzureCostManagement

ExtensionDataSourcePath  
Enrollment Number;1111111



Authentication method  
OAuth2

Privacy level setting for this data source [Learn more](#)  
None

[Go to the app documentation ↗](#) Back Sign in and continue Cancel

11. The final dialog connects to Azure and gets data. *Leave the default values as configured* and select **Sign in and continue**.



## Connect to Azure Cost Management App

X

You are connecting to (2 of 2)

Url

<https://ccmstorageprod.blob.core.windows.net/cc>



Authentication method

Anonymous



Privacy level setting for this data source [Learn more](#)

None



[Go to the app documentation ↗](#)

Back

Sign in and connect

Cancel



12. You are prompted to authenticate with your EA enrollment. Authenticate with Power BI. After you're authenticated, a Power BI data refresh starts.

### ⓘ Note

The data refresh process might take quite a while to complete. The length depends on the number of months specified and the amount of data needed to sync.

After the data refresh is complete, select the Cost Management App to view the pre-created reports.

## Reports available with the app

The following reports are available in the app.

**Getting Started** - Provides useful links to documentation and links to provide feedback.

**Account overview** - The report shows the current billing month summary of information, including:

- Charges against credits
- New purchases
- Azure Marketplace charges
- Overages and total charges

The Billing account overview page might show costs that differ from costs shown in the EA portal.

 **Note**

The **Select date range** selector doesn't affect or change overview tiles. Instead, the overview tiles show the costs for the current billing month. This behavior is intentional.

Data shown in the bar graph is determined by the date selection.

Here's how values in the overview tiles are calculated.

- The value shown in the **Charges against credit** tile is calculated as the sum of `adjustments`.
- The value shown in the **Service overage** tile is calculated as the sum of `ServiceOverage`.
- The value shown in the **Billed separately** tile is calculated as the sum of `chargesBilledseparately`.
- The value shown in the **Azure Marketplace** tile is calculated as the sum of `azureMarketplaceServiceCharges`.
- The value shown in the **New purchase amount** tile is calculated as the sum of `newPurchases`.
- The value shown in the **Total charges** tile is calculated as the sum of (`adjustments` + `ServiceOverage` + `chargesBilledseparately` + `azureMarketplaceServiceCharges`).

The EA portal doesn't show the Total charges column. The Power BI template app includes Adjustments, Service Overage, Charges billed separately, and Azure marketplace service charges as Total charges.

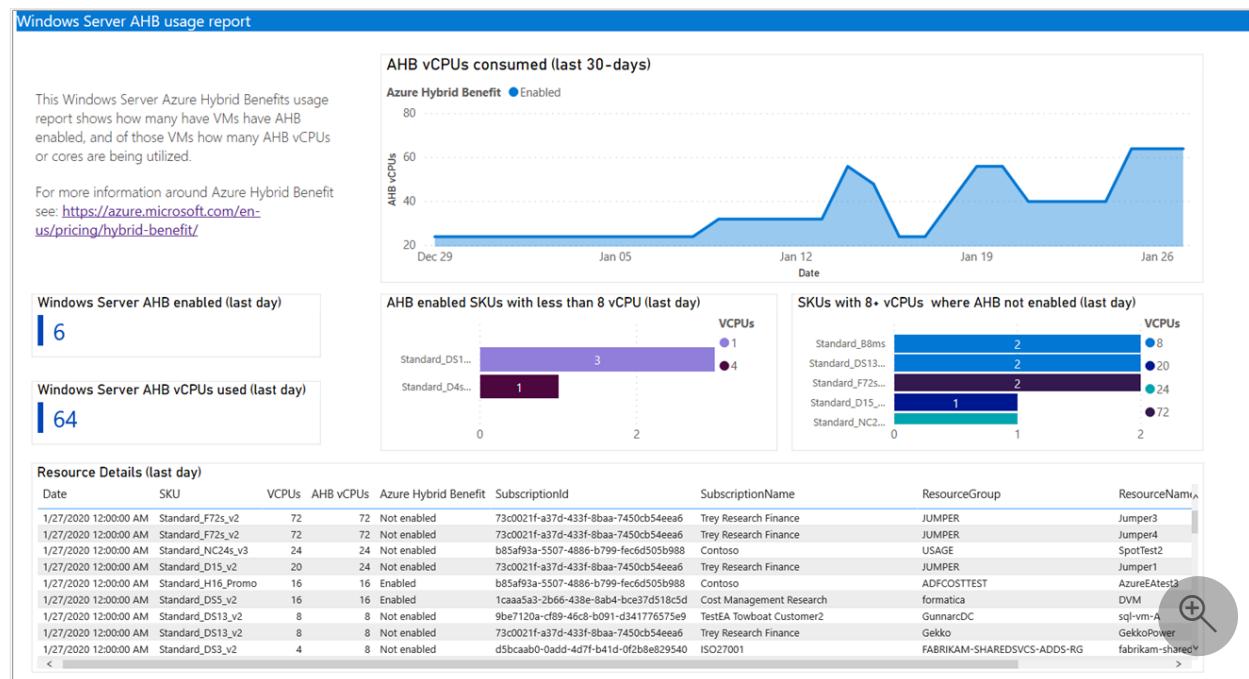
The Prepayment Usage shown in the EA portal isn't available in the Template app as part of the total charges.

**Usage by Subscriptions and Resource Groups** - Provides a cost over time view and charts showing cost by subscription and resource group.

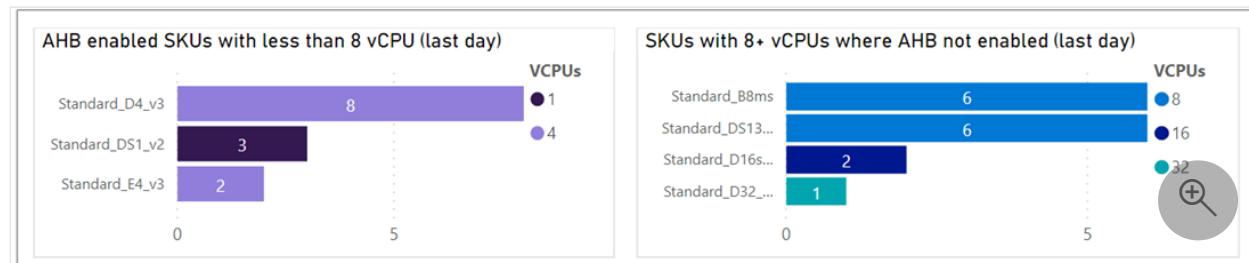
**Usage by Services** - Provides a view over time of usage by MeterCategory. You can track your usage data and drill into any anomalies to understand usage spikes or dips.

**Top 5 Usage drivers** - The report shows a filtered cost summarization by the top 5 MeterCategory and corresponding MeterName.

**Windows Server AHB Usage** - The report shows the number virtual machines that have Azure Hybrid Benefit enabled. It also shows a count of cores/vCPUs used by the virtual machines.



The report also identifies Windows VMs where Hybrid Benefit is **enabled** but there are *less than 8 vCPUs*. It also shows where Hybrid Benefit is **not enabled** that have *8 or more vCPUs*. This information helps you fully use your Hybrid Benefit. Apply the benefit to your most expensive virtual machines to maximize your potential savings.



**RI Chargeback** - The report helps you understand where and how much of a reserved instance (RI) benefit is applied per region, subscription, resource group, or resource. The report uses amortized usage data to show the view.

You can apply a filter on *chargetype* to view RI underutilization data.

For more information about amortized data, see [Get Enterprise Agreement reservation costs and usage](#).

**RI Savings** - The report shows the savings accrued by reservations for subscription, resource group, and the resource level. It displays:

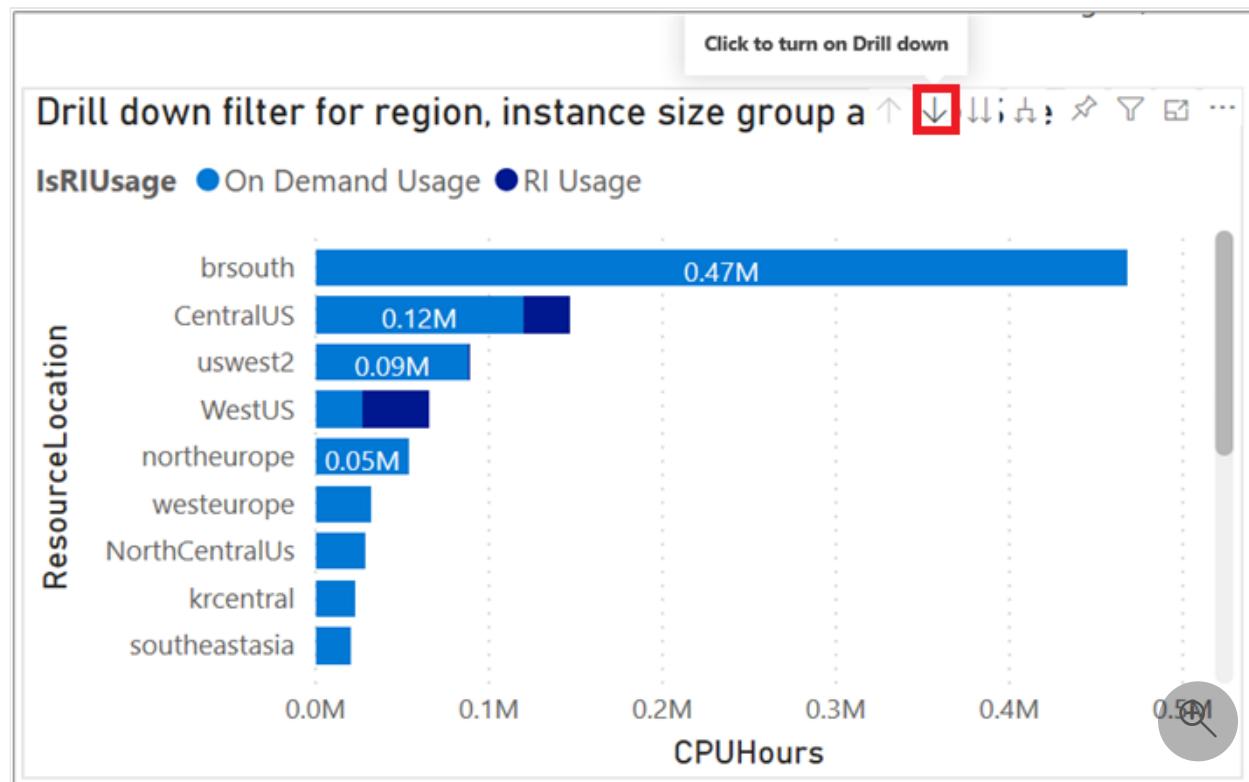
- Cost with reservation
- Estimated on-demand cost if the reservation didn't apply to the usage
- Cost savings accrued from the reservation

The report subtracts any under-utilized reservation waste cost from the total savings. The waste wouldn't occur without a reservation.

You can use the amortized usage data to build on the data.

**VM RI Coverage (shared recommendation)** - The report is split between on-demand VM usage and RI VM usage over the selected period. It provides recommendations for VM RI purchases at a shared scope.

To use the report, select the drill-down filter.



Select the region that you want to analyze. Then select the instance size flexibility group, and so on.

For each drill-down level, the following filters are applied to the report:

- The coverage data on the right is the filter showing how much usage is charged using the on-demand rate vs. how much is covered by the reservation.

- Recommendations are also filtered.

The recommendations table provides recommendations for the reservation purchase, based on the VM sizes used.

The *Normalized Size* and *Recommended Quantity Normalized* values help you normalize the purchase to the smallest size for an instance size flexibility group. The information is helpful if you plan to purchase just one reservation for all sizes in the instance size flexibility group.

Recommendations for shared scope. These recommendations are calculated by evaluating your last 7 days of usage.							
Instance Flexibility Group	Location	SKU	Recommended Quantity	Normalized Size	Recommended Quantity Normalized	Term	Scope
BS Series	centralus	Standard_B2s	1.00	Standard_B1s	8	P3Y	Shared
DSv2 Series	centralus	Standard_DS1_v2	1.00	Standard_DS1_v2	1	P3Y	Shared
DSv2 Series	centralus	Standard_DS2_v2	2.00	Standard_DS1_v2	4	P3Y	Shared
Dv2 Series	centralus	Standard_D1_v2	2.00	Standard_D1_v2	2	P3Y	Shared
Dv2 Series	centralus	Standard_D3_v2	2.00	Standard_D1_v2	8	P3Y	Shared
Dv2 Series High Memory	centralus	Standard_D14_v2	2.00	Standard_D11_v2	16	P3Y	Shared
<b>Total</b>			<b>10.00</b>				

**VM RI Coverage (single recommendation)** - The report is split between on-demand VM usage and RI VM usage over the selected time period. It provides recommendations for VM RI purchases at a subscription scope.

For details about how to use the report, see the [VM RI Coverage \(shared recommendation\)](#) section.

**RI purchases** - The report shows RI purchases over the specified period.

**Price sheet** - The report shows a detailed list of prices specific to a Billing account or EA enrollment.

## Troubleshoot problems

If you're having issues with the Power BI app, the following troubleshooting information might help.

## Error processing the data in the dataset

You might get an error stating:

```
There was an error when processing the data in the dataset.
Data source error: {"error":
{"code":"ModelRefresh_ShortMessage_ProcessingError","pbi.error":
{"code":"ModelRefresh_ShortMessage_ProcessingError","parameters":{},
"details":[{"code":"Message","detail":{"type":1,"value":"We cannot
process the data in the dataset."}}]}}
```

```
convert the value \"Required Field: 'Enr...\" to type
List."}}],"exceptionCulprit":1}} Table: <TableName>.
```

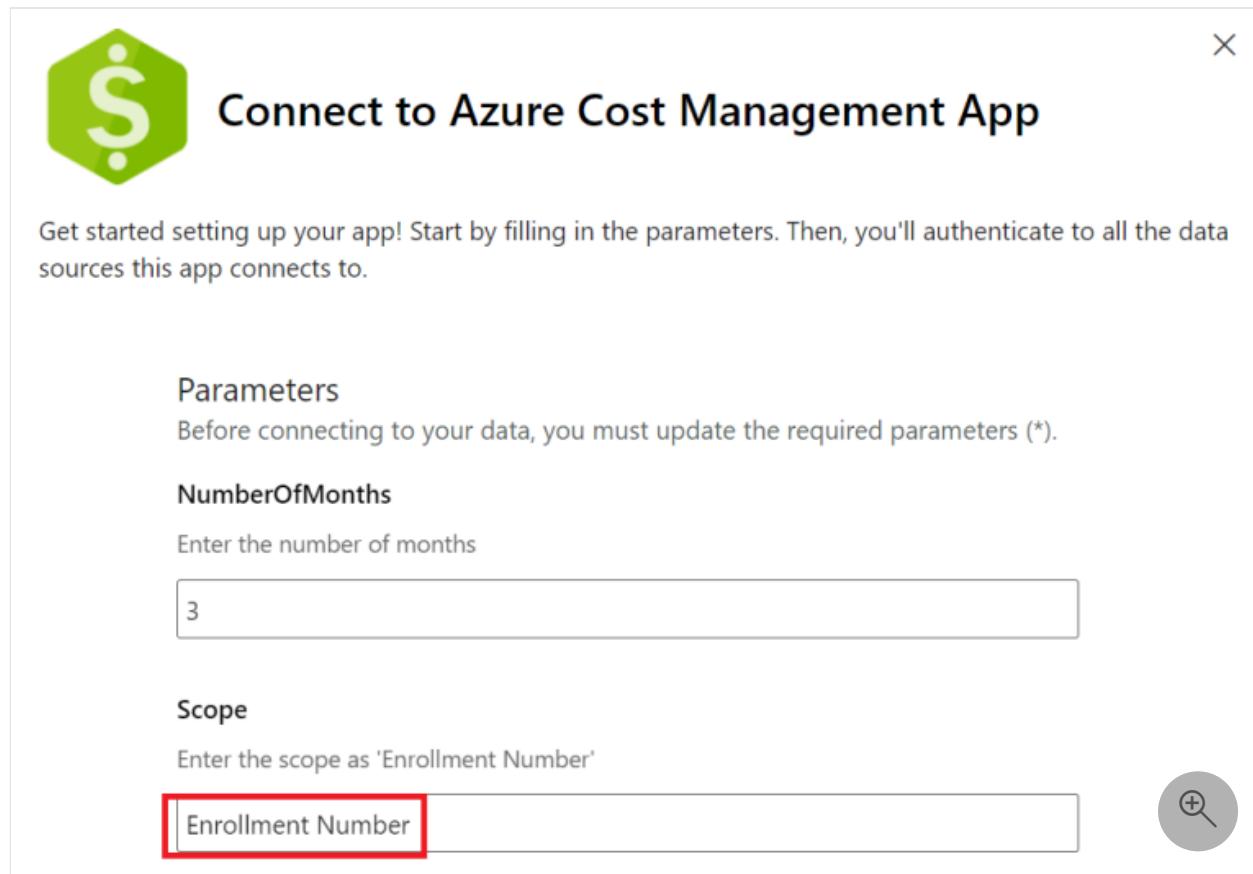
A table name would appear instead of <TableName>.

## Cause

The default **Scope** value of Enrollment Number was changed in the connection to Cost Management.

## Solution

Reconnect to Cost Management and set the **Scope** value to Enrollment Number. Do not enter your organization's enrollment number, instead type Enrollment Number exactly as it appears in the following image.



## BudgetAmount error

You might get an error stating:

Something went wrong  
There was an error when processing the data in the dataset.  
Please try again later or contact support. If you contact support, please provide these details.  
Data source error: The 'budgetAmount' column does not exist in the rowset.  
Table: Budgets.

## Cause

This error occurs because of a bug with the underlying metadata. The issue happens because there's no budget available under **Cost Management > Budget** in the Azure portal. The bug fix is in the process of getting deployed to the Power BI Desktop and Power BI service.

## Solution

- Until the bug is fixed, you can work around the problem by adding a test budget in the Azure portal at the billing account/EA enrollment level. The test budget unblocks connecting with Power BI. For more information about creating a budget, see [Tutorial: Create and manage budgets](#).

## Invalid credentials for AzureBlob error

You might get an error stating:

Failed to update data source credentials: The credentials provided for the AzureBlobs source are invalid.

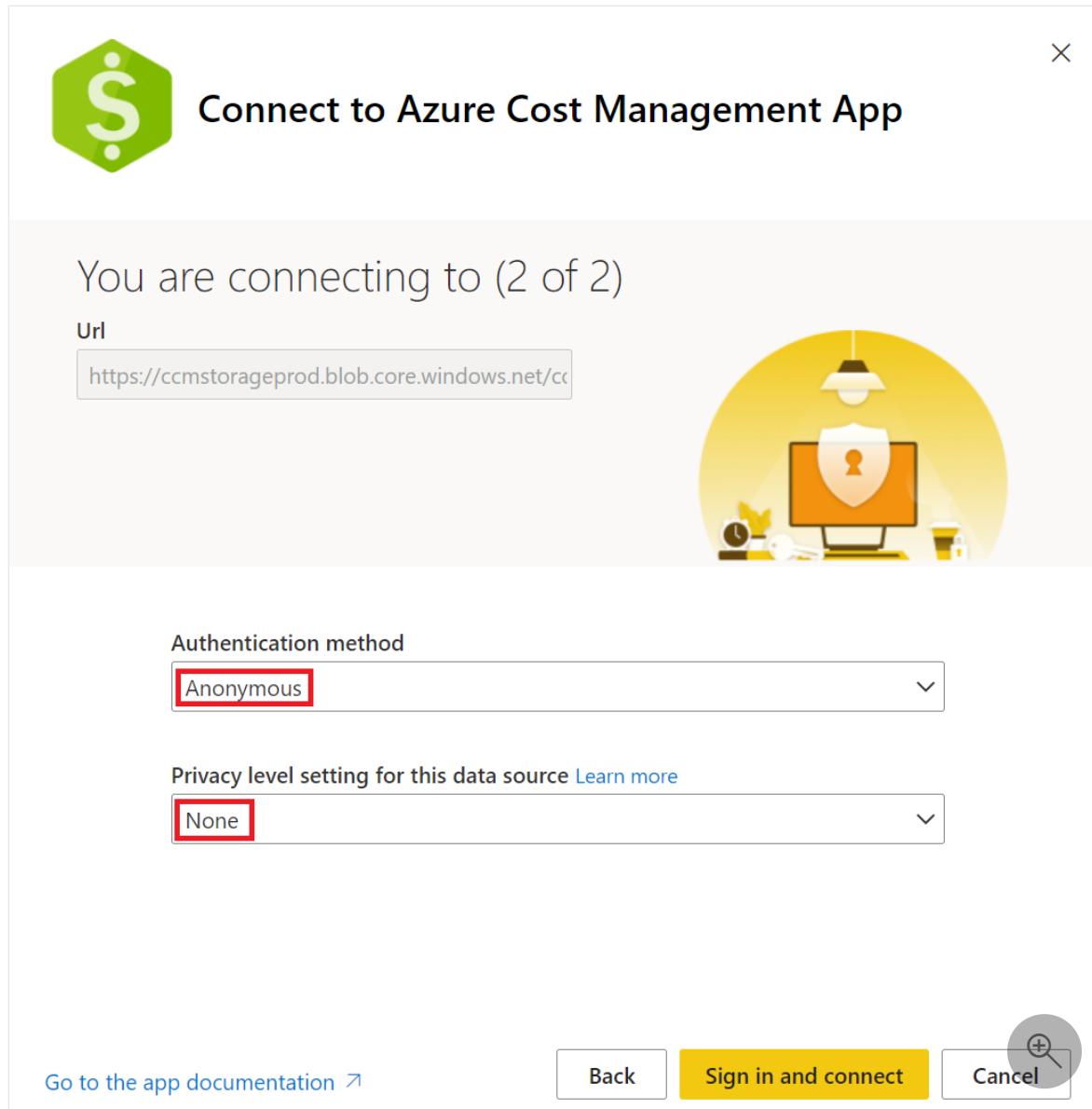
## Cause

This error occurs if you change the authentication method for your data source connection.

## Solution

1. Connect to your data.
2. After you enter your EA enrollment and number of months, make sure that you leave the default value of **Anonymous** for Authentication method and **None** for

the Privacy level setting.



You are connecting to (2 of 2)

Url

https://ccmstorageprod.blob.core.windows.net/cc

Authentication method

Anonymous

Privacy level setting for this data source [Learn more](#)

None

Go to the app documentation ↗

Back

Sign in and connect

Cancel

3. On the next page, set OAuth2 for the Authentication method and None set for Privacy level. Then, sign in to authenticate with your enrollment. This step also starts a Power BI data refresh.

## Data reference

The following information summarizes the data available through the app. There's also links to APIs that give in-depth details for data fields and values.

Table reference	Description
AutoFitComboMeter	Data included in the app to normalize the RI recommendation and usage to the smallest size in the instance family group.
Balance summary	Summary of the balance for Enterprise Agreements.

Table reference	Description
<a href="#">Budgets</a>	Budget details to view actual costs or usage against existing budget targets.
<a href="#">Pricesheets</a>	Applicable meter rates for the provided billing profile or EA enrollment.
<a href="#">RI charges</a>	Charges associated to your reserved instances over the last 24 months.
<a href="#">RI recommendations (shared)</a>	Reserved instance purchase recommendations based on all your subscription usage trends for the last 7 days.
<a href="#">RI recommendations (single)</a>	Reserved instance purchase recommendations based on your single subscription usage trends for the last 7 days.
<a href="#">RI usage details</a>	Consumption details for your existing reserved instances over the last month.
<a href="#">RI usage summary</a>	Daily Azure reservation usage percentage.
<a href="#">Usage details</a>	A breakdown of consumed quantities and estimated charges for the given billing profile in the EA enrollment.
<a href="#">Usage details amortized</a>	A breakdown of consumed quantities and estimated amortized charges for the given billing profile in the EA enrollment.

## Next steps

For more information about configuring data, refresh, sharing reports, and additional report customization see the following articles:

- [Configure scheduled refresh](#)
- [Share Power BI dashboards and reports with coworkers and others](#)
- [Subscribe yourself and others to reports and dashboards in the Power BI service](#)
- [Download a report from the Power BI service to Power BI Desktop](#)
- [Save a report in Power BI service and Power BI Desktop](#)
- [Create a report in the Power BI service by importing a dataset](#)

# Group and filter options in Cost analysis and budgets

Article • 03/07/2023

Cost analysis has many grouping and filtering options. This article helps you understand when to use them.

To watch a video about grouping and filtering options, watch the [Cost Management reporting by dimensions and tags](#) video. To watch other videos, visit the [Cost Management YouTube channel](#).

<https://www.youtube-nocookie.com/embed/2Vx7V17zmk>

## Group and filter properties

The following table lists some of the most common grouping and filtering options available in Cost analysis and budgets. See the notes column to learn when to use them.

Some filters are only available to specific offers. For example, a billing profile isn't available for an enterprise agreement. For more information, see [Supported Microsoft Azure offers](#).

Property	When to use	Notes
Availability zones	Break down AWS costs by availability zone.	Applicable only to AWS scopes and management groups. Azure data doesn't include availability zone and will show as <b>No availability zone</b> .
Billing period	Break down PAYG costs by the month that they were, or will be, invoiced.	Use <b>Billing period</b> to get a precise representation of invoiced PAYG charges. Include two extra days before and after the billing period if filtering down to a custom date range. Limiting to the exact billing period dates won't match the invoice. Will show costs from all invoices in the billing period. Use <b>Invoice ID</b> to filter down to a specific invoice. Applicable only to PAYG subscriptions because EA and MCA are billed by calendar months. EA/MCA accounts can use calendar months in the date picker or monthly granularity to accomplish the same goal.

Property	When to use	Notes
<b>BillingProfileId</b>	The ID of the billing profile that is billed for the subscription's charges.	Unique identifier of the EA enrollment, pay-as-you-go subscription, MCA billing profile, or AWS consolidated account.
<b>BillingProfileName</b>	Name of the EA enrollment, pay-as-you-go subscription, MCA billing profile, or AWS consolidated account.	Name of the EA enrollment, pay-as-you-go subscription, MCA billing profile, or AWS consolidated account.
<b>Charge type</b>	Break down usage, purchase, refund, and unused reservation and savings plan costs.	Reservation purchases, savings plan purchases, and refunds are available only when using actual costs and not when using amortized costs. Unused reservation and savings plan costs are available only when looking at amortized costs.
<b>Department</b>	Break down costs by EA department.	Available only for EA and management groups. PAYG subscriptions don't have a department and will show as <b>No department or unassigned</b> .
<b>Enrollment account</b>	Break down costs by EA account owner.	Available only for EA billing accounts, departments, and management groups. PAYG subscriptions don't have EA enrollment accounts and will show as <b>No enrollment account or unassigned</b> .
<b>Frequency</b>	Break down usage-based, one-time, and recurring costs.	Indicates whether a charge is expected to repeat. Charges can either happen once <b>OneTime</b> , repeat on a monthly or yearly basis <b>Recurring</b> , or be based on usage <b>UsageBased</b> .
<b>Invoice ID</b>	Break down costs by billed invoice.	Unbilled charges don't have an invoice ID yet and EA costs don't include invoice details and will show as <b>No invoice ID</b> .
<b>InvoiceSectionId</b>	Unique identifier for the MCA invoice section.	Unique identifier for the EA department or MCA invoice section.
<b>InvoiceSectionName</b>	Name of the invoice section.	Name of the EA department or MCA invoice section.

Property	When to use	Notes
Location	Break down costs by resource location or region.	Purchases and Marketplace usage may be shown as <b>unassigned</b> , or <b>No resource location</b> .
Meter	Break down costs by usage meter.	Purchases and Marketplace usage will show as <b>unassigned</b> or <b>No meter</b> . Refer to <b>Charge type</b> to identify purchases and <b>Publisher type</b> to identify Marketplace charges.
Operation	Break down AWS costs by operation.	Applicable only to AWS scopes and management groups. Azure data doesn't include operation and will show as <b>No operation</b> - use <b>Meter</b> instead.
Pricing model	Break down costs by on-demand, reservation, or spot usage.	Purchases show as <b>OnDemand</b> . If you see <b>Not applicable</b> , group by <b>Reservation</b> to determine whether the usage is reservation or on-demand usage and <b>Charge type</b> to identify purchases.
PartNumber	The identifier used to get specific meter pricing.	
Product	Name of the product.	
ProductOrderId	Unique identifier for the product order	
ProductOrderName	Unique name for the product order.	
Provider	Break down costs by the provider type: Azure, Microsoft 365, Dynamics 365, AWS, and so on.	Identifier for product and line of business.

Property	When to use	Notes
Publisher type	Break down Microsoft, Azure, AWS, and Marketplace costs.	Values are <b>Microsoft</b> for MCA accounts and <b>Azure</b> for EA and pay-as-you-go accounts.
Reservation	Break down costs by reservation.	Any usage or purchases that aren't associated with a reservation will show as <b>No reservation</b> or <b>No values</b> . Group by <b>Publisher type</b> to identify other Azure, AWS, or Marketplace purchases.
ReservationId	Unique identifier for the purchased reservation instance.	In actual costs, use ReservationID to know which reservation the charge is for.
ReservationName	Name of the purchased reservation instance.	In actual costs, use ReservationName to know which reservation the charge is for.
Resource	Break down costs by resource.	Marketplace purchases show as <b>Other Marketplace purchases</b> and Azure purchases, like Reservations and Support charges, show as <b>Other Azure purchases</b> . Group by or filter on <b>Publisher type</b> to identify other Azure, AWS, or Marketplace purchases.
Resource group	Break down costs by resource group.	Purchases, tenant resources not associated with subscriptions, subscription resources not deployed to a resource group, and classic resources don't have a resource group and will show as <b>Other Marketplace purchases</b> , <b>Other Azure purchases</b> , <b>Other tenant resources</b> , <b>Other subscription resources</b> , <b>\$system</b> , or <b>Other charges</b> .
ResourceId	Unique identifier of the <a href="#">Azure Resource Manager</a> resource.	
Resource type	Break down costs by resource type.	Type of resource instance. Not all charges come from deployed resources. Charges that don't have a resource type will be shown as null or empty, <b>Others</b> , or <b>Not applicable</b> . For example, purchases and classic services will show as <b>others</b> , <b>classic services</b> , or <b>No resource type</b> .

Property	When to use	Notes
ServiceFamily	Type of Azure service. For example, Compute, Analytics, and Security.	
ServiceName	Name of the Azure service.	Name of the classification category for the meter. For example, Cloud services and Networking.
Service name or Meter category	Break down cost by Azure service.	Purchases and Marketplace usage will show as <b>No service name or unassigned</b> .
Service tier or Meter subcategory	Break down cost by Azure usage meter subclassification.	Purchases and Marketplace usage will be empty or show as <b>unassigned</b> .
Subscription	Break down costs by Azure subscription and AWS linked account.	Purchases and tenant resources may show as <b>No subscription</b> .
Tag	Break down costs by tag values for a specific tag key.	Purchases, tenant resources not associated with subscriptions, subscription resources not deployed to a resource group, and classic resources cannot be tagged and will show as <b>Tags not supported</b> . Services that don't include tags in usage data will show as <b>Tags not available</b> . Any remaining cases where tags aren't specified on a resource will show as <b>Untagged</b> . Learn more about <a href="#">tags support for each resource type</a> .
UnitOfMeasure	The billing unit of measure for the service. For example, compute services are billed per hour.	

For more information about terms, see [Understand the terms used in the Azure usage and charges file](#).

## Publisher Type value changes

In Cost Management, the `PublisherType` field indicates whether charges are for Microsoft, Marketplace, or AWS (if you have a [Cross Cloud connector](#) configured) products.

## What changed?

Effective 14 October 2021, the `PublisherType` field with the value `Azure` was updated to `Microsoft` for all customers with a [Microsoft Customer Agreement](#). The change was made to accommodate enhancements to support Microsoft products other than Azure like Microsoft 365 and Dynamics 365.

Values of `Marketplace` and `AWS` remain unchanged.

The change didn't affect customers with an Enterprise Agreement or pay-as-you-go offers.

## Impact and action

For any Cost Management data that you've downloaded before 14 October 2021, consider the `PublisherType` change from the older `Azure` and the new `Microsoft` field values. The data could have been downloaded through exports, usage details, or from Cost Management.

If you use Cost Management + Billing REST API calls that filter the `PublisherType` field by the value `Azure`, you need to address the change and filter by the new value `Microsoft` after 14 October 2021. If you make any API calls with a filter for Publisher type = `Azure`, data won't be returned.

There's no impact to Cost analysis or budgets because the changes are automatically reflected in the filters. Any saved views or budgets created with Publisher Type = "Azure" filter will be automatically updated.

## Next steps

- [Start analyzing costs.](#)

# What are Azure Reservations?

Article • 04/17/2023

Azure Reservations help you save money by committing to one-year or three-year plans for multiple products. Committing allows you to get a discount on the resources you use. Reservations can significantly reduce your resource costs by up to 72% from pay-as-you-go prices. Reservations provide a billing discount and don't affect the runtime state of your resources. After you purchase a reservation, the discount automatically applies to matching resources.

You can pay for a reservation up front or monthly. The total cost of up-front and monthly reservations is the same and you don't pay any extra fees when you choose to pay monthly. Monthly payment is available for Azure reservations, not third-party products.

You can buy a reservation in the [Azure portal](#).

## Why buy a reservation?

If you have consistent resource usage that supports reservations, buying a reservation gives you the option to reduce your costs. For example, when you continuously run instances of a service without a reservation, you're charged at pay-as-you-go rates. When you buy a reservation, you immediately get the reservation discount. The resources are no longer charged at the pay-as-you-go rates.

## How reservation discount is applied

After purchase, the reservation discount automatically applies to the resource usage that matches the attributes you select when you buy the reservation. Attributes include the SKU, regions (where applicable), and scope. Reservation scope selects where the reservation savings apply.

For more information about how discount is applied, see [Reserved instance discount application](#).

For more information about how reservation scope works, see [Scope reservations](#).

## Determine what to purchase

All reservations, except Azure Databricks, are applied on an hourly basis. Consider reservation purchases based on your consistent base usage. You can determine which reservation to purchase by analyzing your usage data or by using reservation recommendations. Recommendations are available in:

- Azure Advisor (VMs only)
- Reservation purchase experience in the Azure portal
- Cost Management Power BI app
- APIs

For more information, see [Determine what reservation to purchase](#)

## Buying a reservation

You can purchase reservations from the Azure portal, APIs, PowerShell, and CLI.

Go to the [Azure portal](#) ↗ to make a purchase.

For more information, see [Buy a reservation](#).

## How is a reservation billed?

The reservation is charged to the payment method tied to the subscription. The reservation cost is deducted from your Azure Prepayment (previously called monetary commitment) balance, if available. When your Azure Prepayment balance doesn't cover the cost of the reservation, you're billed the overage. If you have a subscription from an individual plan with pay-as-you-go rates, the credit card you have on your account is billed immediately for up-front purchases. Monthly payments appear on your invoice and your credit card is charged monthly. When you're billed by invoice, you see the charges on your next invoice.

## Who can manage a reservation by default

By default, the following users can view and manage reservations:

- The person who buys a reservation and the account administrator of the billing subscription used to buy the reservation are added to the reservation order.
- Enterprise Agreement and Microsoft Customer Agreement billing administrators.

To allow other people to manage reservations, see [Manage Reservations for Azure resources](#).

# Get reservation details and utilization after purchase

If you have permission to view to the reservation, you can see it and its use in the Azure portal. You can get the data using APIs, as well.

For more information on how to see reservations in Azure portal, see [View reservations in the Azure portal](#)

## Manage reservations after purchase

After you buy an Azure reservation, you can update the scope to apply reservation to a different subscription, change who can manage the reservation, split a reservation into smaller parts, or change instance size flexibility.

For more information, see [Manage Reservations for Azure resources](#)

## Flexibility with Azure reservations

Azure Reservations provide flexibility to help meet your evolving needs. You can exchange a reservation for another reservation of the same type. You can also refund a reservation, up to \$50,000 USD in a 12 month rolling window, if you no longer need it. The maximum limit of the refund applies to all reservations in the scope of your agreement with Microsoft.

For more information, see [Self-service exchanges and refunds for Azure Reservations](#)

## Charges covered by reservation

- **Reserved Virtual Machine Instance** - A reservation only covers the virtual machine and cloud services compute costs. It doesn't cover additional software, Windows, networking, or storage charges.
- **Azure Blob storage reserved capacity** - A reservation covers storage capacity for Blob storage and Azure Data Lake Gen2 storage. The reservation doesn't cover bandwidth or transaction rates.
- **Azure Files reserved capacity** - A reservation covers storage capacity for Azure Files. Reservations for hot and cool tiers don't cover bandwidth or transaction rates.
- **Azure Cosmos DB reserved capacity** - A reservation covers throughput provisioned for your resources. It doesn't cover the storage and networking

charges.

- **Azure Data Factory data flows** - A reservation covers integration runtime cost for the compute type and number of cores that you buy.
- **SQL Database reserved vCore** - Covers both SQL Managed Instance and SQL Database Elastic Pool/single database. Only the compute costs are included with a reservation. The SQL license is billed separately.
- **Azure Synapse Analytics** - A reservation covers cDWU usage. It doesn't cover storage or networking charges associated with the Azure Synapse Analytics usage.
- **Azure Databricks** - A reservation covers only the DBU usage. Other charges, such as compute, storage, and networking, are applied separately.
- **App Service stamp fee** - A reservation covers stamp usage. It doesn't apply to workers, so any other resources associated with the stamp are charged separately.
- **Azure Database for MySQL** - Only the compute costs are included with a reservation. A reservation doesn't cover software, networking, or storage charges associated with the MySQL Database server.
- **Azure Database for PostgreSQL** - Only the compute costs are included with a reservation. A reservation doesn't cover software, networking, or storage charges associated with the PostgreSQL Database servers.
- **Azure Database for MariaDB** - Only the compute costs are included with a reservation. A reservation doesn't cover software, networking, or storage charges associated with the MariaDB Database server.
- **Azure Data Explorer** - A reservation covers the markup charges. A reservation doesn't apply to compute, networking, or storage charges associated with the clusters.
- **Azure Cache for Redis** - Only the compute costs are included with a reservation. A reservation doesn't cover networking or storage charges associated with the Redis cache instances.
- **Azure Dedicated Host** - Only the compute costs are included with the Dedicated host.
- **Azure Disk Storage reservations** - A reservation only covers premium SSDs of P30 size or greater. It doesn't cover any other disk types or sizes smaller than P30.
- **Azure Backup Storage reserved capacity** - A capacity reservation lowers storage costs of backup data in a Recovery Services Vault.

Software plans:

- **SUSE Linux** - A reservation covers the software plan costs. The discounts apply only to SUSE meters and not to the virtual machine usage.
- **Red Hat Plans** - A reservation covers the software plan costs. The discounts apply only to RedHat meters and not to the virtual machine usage.

- **Azure Red Hat OpenShift** - A reservation applies to the OpenShift costs, not to Azure infrastructure costs.

For Windows virtual machines and SQL Database, the reservation discount doesn't apply to the software costs. You can cover the licensing costs with [Azure Hybrid Benefit](#).

## Need help? Contact us.

If you have questions or need help, [create a support request](#).

## Next steps

- Learn more about Azure Reservations with the following articles:
  - [Manage Azure Reservations](#)
  - [Understand reservation usage for your subscription with pay-as-you-go rates](#)
  - [Understand reservation usage for your Enterprise enrollment](#)
  - [Windows software costs not included with reservations](#)
  - [Azure Reservations in Partner Center Cloud Solution Provider \(CSP\) program](#)
- Learn more about reservations for service plans:
  - [Virtual Machines with Azure Reserved VM Instances](#)
  - [Azure Cosmos DB resources with Azure Cosmos DB reserved capacity](#)
  - [SQL Database compute resources with Azure SQL Database reserved capacity](#)
  - [Azure Cache for Redis resources with Azure Cache for Redis reserved capacity](#)  
Learn more about reservations for software plans:
    - [Red Hat software plans from Azure Reservations](#)
    - [SUSE software plans from Azure Reservations](#)

# How a reservation discount is applied

Article • 04/20/2023

This article helps you understand how reserved instance discounts are generally applied. The reservation discount applies to the resource usage matching the attributes you select when you buy the reservation. Attributes include the scope where the matching VMs, SQL databases, Azure Cosmos DB, or other resources run. For example, if you want a reservation discount for four Standard D2 virtual machines in the West US region, then select the subscription where the VMs are running.

A reservation discount is "*use-it-or-lose-it*". If you don't have matching resources for any hour, then you lose a reservation quantity for that hour. You can't carry forward unused reserved hours.

When you shut down a resource, the reservation discount automatically applies to another matching resource in the specified scope. If no matching resources are found in the specified scope, then the reserved hours are *lost*.

For example, you might later create a resource and have a matching reservation that is underutilized. The reservation discount automatically applies to the new matching resource.

If the virtual machines are running in different subscriptions within your enrollment/account, then select the scope as shared. Shared scope allows the reservation discount to be applied across subscriptions. You can change the scope after you buy a reservation. For more information, see [Manage Azure Reservations](#). You can also use the management group scope. It applies the reservation discount to the matching resource in the list of subscriptions that are a part of both the management group and billing scope.

A reservation discount only applies to resources associated with Enterprise, Microsoft Customer Agreement, CSP, or subscriptions with pay-as-you go rates. Resources that run in a subscription with other offer types don't receive the reservation discount.

## When the reservation term expires

At the end of the reservation term, the billing discount expires, and the resources are billed at the pay-as-you go price. By default, the reservations are not set to renew automatically. You can choose to enable automatic renewal of a reservation by selecting the option in the renewal settings. With automatic renewal, a replacement reservation will be purchased upon expiry of the existing reservation. By default, the replacement

reservation has the same attributes as the expiring reservation, optionally you change the billing frequency, term, or quantity in the renewal settings. Any user with owner access on the reservation and the subscription used for billing can set up renewal.

## Discount applies to different sizes

When you buy a reservation, the discount can apply to other instances with attributes that are within the same size group. This feature is known as instance size flexibility. The flexibility of the discount coverage depends on the type of reservation and the attributes you pick when you buy the reservation.

Service plans:

- Reserved VM Instances: When you buy the reservation and select **Optimized for instance size flexibility**, the discount coverage depends on the VM size you select. The reservation can apply to the virtual machines (VMs) sizes in the same size series group. For more information, see [Virtual machine size flexibility with Reserved VM Instances](#).
- Azure Storage reserved capacity: You can purchase reserved capacity for standard Azure Storage accounts in units of 100 TiB or 1 PiB per month. For information about which regions support Azure Storage reserved capacity, see [Block blob pricing](#). Azure Storage reserved capacity is available for all access tiers (hot, cool, and archive) and for any replication configuration (LRS, GRS, or ZRS).
- SQL Database reserved capacity: The discount coverage depends on the performance tier you pick. For more information, see [Understand how an Azure reservation discount is applied](#).
- Azure Cosmos DB reserved capacity: The discount coverage depends on the provisioned throughput. For more information, see [Understand how an Azure Cosmos DB reservation discount is applied](#).

## How discounts apply to specific Azure services

Read the following articles that apply to you to learn how discounts apply to a specific Azure service:

- [App Service](#)
- [Azure Cache for Redis](#)
- [Azure Cosmos DB](#)
- [Azure SQL Edge](#)
- [Database for MariaDB](#)
- [Database for MySQL](#)

- Database for PostgreSQL
- Databricks
- Data Explorer
- Dedicated Hosts
- Disk Storage
- Red Hat Linux Enterprise
- Software plans
- Storage
- SQL Database
- Azure Synapse Analytics
- Virtual machines

## Next steps

- Manage Azure Reservations
- Understand reservation usage for your subscription with pay-as-you-go rates
- Understand reservation usage for your Enterprise enrollment
- Windows software costs not included with reservations

# Get Enterprise Agreement and Microsoft Customer Agreement reservation costs and usage

Article • 12/07/2022

Enhanced data for reservation costs and usage is available for Enterprise Agreement (EA) and Microsoft Customer Agreement (MCA) usage in Cost management. This article helps you:

- Get reservation purchase data
- Know which subscription, resource group or resource used the reservation
- Chargeback for reservation utilization
- Calculate reservation savings
- Get reservation under-utilization data
- Amortize reservation costs

Marketplace charges are consolidated in usage data. You view charges for first party usage, marketplace usage, and purchases from a single data source.

## Reservation charges in Azure usage data

Data is divided into two separate data sets: *Actual Cost* and *Amortized Cost*. How these two datasets differ:

**Actual Cost** - Provides data to reconcile with your monthly bill. This data has reservation purchase costs and reservation application details. With this data, you can know which subscription or resource group or resource received the reservation discount in a particular day. The EffectivePrice for the usage that receives the reservation discount is zero.

**Amortized Cost** - This dataset is similar to the Actual Cost dataset except that - the EffectivePrice for the usage that gets reservation discount is the prorated cost of the reservation (instead of being zero). This helps you know the monetary value of reservation consumption by a subscription, resource group or a resource, and can help you charge back for the reservation utilization internally. The dataset also has unused reservation hours. The dataset does not have reservation purchase records.

Comparison of two data sets:

Data	Actual Cost data set	Amortized Cost data set
Reservation purchases	<p>Available in this view.</p> <p>To get this data filter on ChargeType = "Purchase".</p> <p>Refer to ReservationID or ReservationName to know which reservation the charge is for.</p>	Not applicable to this view.
EffectivePrice	The value is zero for usage that gets reservation discount.	The value is per-hour prorated cost of the reservation for usage that has the reservation discount.
Unused reservation (Provides the number of hours the reservation wasn't used in a day and the monetary value of the waste)	<p>Not applicable in this view.</p> <p>To get this data, filter on ChargeType = "UnusedReservation".</p> <p>Refer to ReservationID or ReservationName to know which reservation was underutilized. This is how much of the reservation was wasted in for the day.</p>	<p>Available in this view.</p>
UnitPrice(Price of the resource from your price sheet)	Available	Available

Other information available in Azure usage data has changed:

- Product and Meter information - Azure doesn't replace the originally consumed meter with the ReservationId and ReservationName, as it did previously.
- ReservationId and ReservationName - They are their own fields in the data. Previously, it used to be available only under AdditionalInfo.
- ProductOrderId - The reservation order ID, added as its own field.
- ProductOrderName - The product name of the purchased reservation.
- Term - 12 months or 36 months.
- RINormalizationRatio - Available under AdditionalInfo. This is the ratio where the reservation is applied to the usage record. If instance size flexibility is enabled on for your reservation, then it can apply to other sizes. The value shows the ratio that the reservation was applied to for the usage record.

For more information, see the Usage details field [Definitions](#).

# Get Azure consumption and reservation usage data using API

You can get the data using the API or download it from Azure portal.

For information about permissions needed to view and manage reservations, see [Who can manage a reservation by default](#).

You call the [Usage Details API](#) to get the new data. For details about terminology, see [usage terms](#).

Here's an example call to the Usage Details API:

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{enrollmentId}/providers/Microsoft.Billing/billingPeriods/{billingPeriodId}/providers/Microsoft.Consumption/usagedetails?metric={metric}&api-version=2019-05-01&$filter={filter}
```

For more information about {enrollmentId} and {billingPeriodId}, see the [Usage Details – List API](#) article.

Information in the following table about metric and filter can help solve for common reservation problems.

Type of API data	API call action
All Charges (usage and purchases)	Replace {metric} with ActualCost
Usage that got reservation discount	Replace {metric} with ActualCost Replace {filter} with: properties/reservationId%20ne%20
Usage that didn't get reservation discount	Replace {metric} with ActualCost Replace {filter} with: properties/reservationId%20eq%20
Amortized charges (usage and purchases)	Replace {metric} with AmortizedCost

Type of API data	API call action
Unused reservation report	Replace {metric} with AmortizedCost  Replace {filter} with: properties/ChargeType%20eq%20'UnusedReservation'
Reservation purchases	Replace {metric} with ActualCost  Replace {filter} with: properties/ChargeType%20eq%20'Purchase'
Refunds	Replace {metric} with ActualCost  Replace {filter} with: properties/ChargeType%20eq%20'Refund'

## Download the EA usage CSV file with new data

If you're an EA admin, you can download the CSV file that contains new usage data from Azure portal. This data isn't available from the EA portal (ea.azure.com), you must download the usage file from Azure portal (portal.azure.com) to see the new data.

In the Azure portal, navigate to [Cost management + billing](#).

1. Select the billing account.
2. Select **Usage + charges**.
3. Select **Download**.

MONTH	CHARGES AGAINST CREDITS	SERVICE OVERAGE	BILLED SEPARATELY	AZURE MARKETPLACE	TOTAL CHARGES	DOWNLOAD
Apr 2019	0.00	378,065.28	0.00	524.20	378,589.48	⬇
Mar 2019	1,000.00	1,618,121.87	0.00	17,879.74	1,637,001.61	⬇

4. In **Download Usage + Charges**, under **Usage Details Version 2**, select **All Charges** (usage and purchases) and then select download. Repeat for **Amortized charges** (usage and purchases).

## Download usage for your Microsoft Customer Agreement

To view and download usage data for a billing profile, you must be a billing profile Owner, Contributor, Reader, or Invoice manager.

### Download usage for billed charges

1. Search for **Cost Management + Billing**.
2. Select a billing profile.
3. Select **Invoices**.
4. In the invoice grid, find the row of the invoice corresponding to the usage you want to download.
5. Select the ellipsis ( ...) at the end of the row.
6. In the download context menu, select **Azure usage and charges**.

## Common cost and usage tasks

The following sections are common tasks that most people use to view their reservation cost and usage data.

### Get reservation purchase costs

Reservation purchase costs are available in Actual Cost data. Filter for *ChargeType* = *Purchase*. Refer to *ProductOrderId* to determine which reservation order the purchase is for.

### Get underutilized reservation quantity and costs

Get Amortized Cost data and filter for *ChargeType* = *UnusedReservation*. You get the daily unused reservation quantity and the cost. You can filter the data for a reservation or reservation order using *ReservationId* and *ProductOrderId* fields, respectively. If a reservation was 100% utilized, the record has a quantity of 0.

### Amortize reservation costs

Get Amortized Cost data and filter for a reservation order using *ProductOrderId* to get daily amortized costs for a reservation.

### Chargeback for a reservation

You can chargeback reservation use to other organizations by subscription, resource groups, or tags. Amortized cost data provides monetary value of a reservation's utilization at the following data types:

- Resources (such as a VM)
- Resource group
- Tags

- Subscription

## Get the blended rate for chargeback

To determine the blended rate, get the amortized costs data and aggregate the total cost. For VMs, you can use either MeterName or ServiceType information from AdditionalInfo JSON data. Divide the total cost by the quantity used to get the blended rate.

## Audit optimum reservation use for instance size flexibility

Multiple the quantity with the *RINormalizationRatio*, from AdditionalInfo. The results indicate how many hours of reservation use was applied to the usage record.

## Determine reservation savings

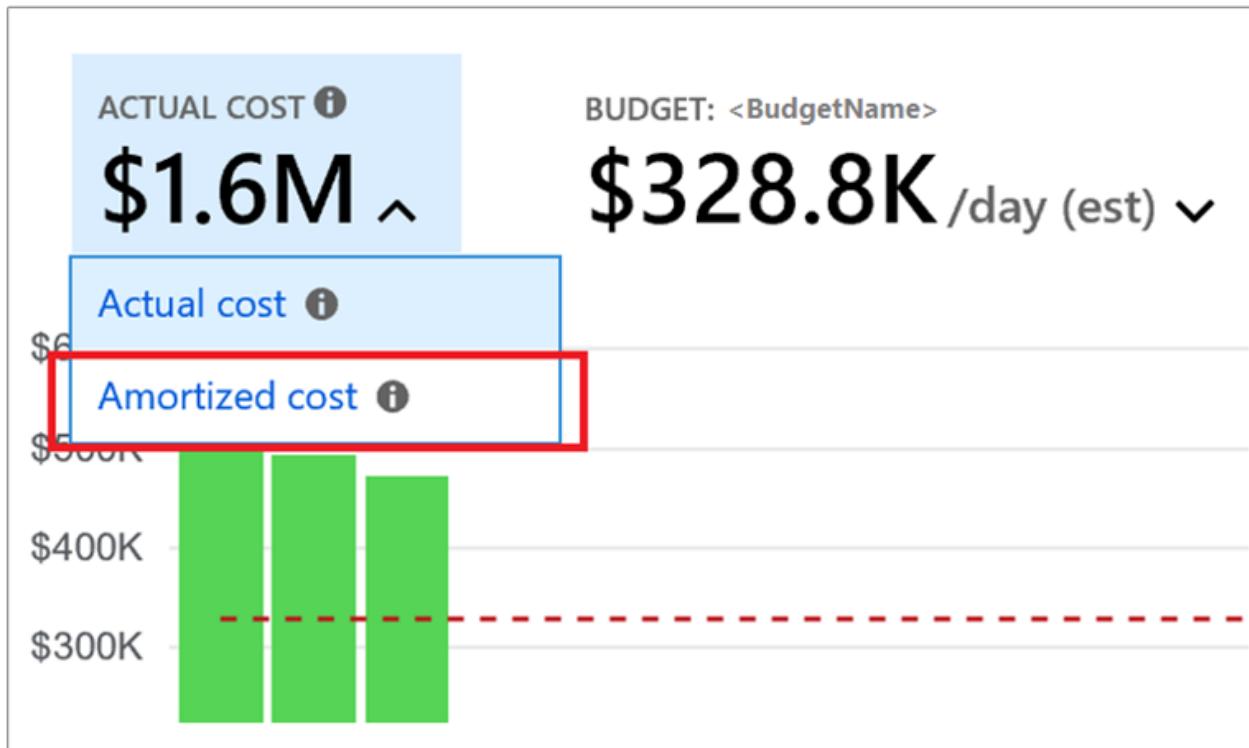
Get the Amortized costs data and filter the data for a reserved instance. Then:

1. Get estimated pay-as-you-go costs. Multiply the *UnitPrice* value with *Quantity* values to get estimated pay-as-you-go costs, if the reservation discount didn't apply to the usage.
2. Get the reservation costs. Sum the *Cost* values to get the monetary value of what you paid for the reserved instance. It includes the used and unused costs of the reservation.
3. Subtract reservation costs from estimated pay-as-you-go costs to get the estimated savings.

Keep in mind that if you have an underutilized reservation, the *UnusedReservation* entry for *ChargeType* becomes a factor to consider. When you have a fully utilized reservation, you receive the maximum savings possible. Any *UnusedReservation* quantity reduces savings.

## Reservation purchases and amortization in cost analysis

Reservation costs are available in [cost analysis](#). By default, cost analysis shows **Actual cost**, which is how costs will be shown on your bill. To view reservation purchases broken down and associated with the resources which used the benefit, switch to **Amortized cost**:



Group by charge type to see a break down of usage, purchases, and refunds; or by reservation for a breakdown of reservation and on-demand costs. Remember the only reservation costs you will see when looking at actual cost are purchases, but costs will be allocated to the individual resources which used the benefit when looking at amortized cost. You will also see a new **UnusedReservation** charge type when looking at amortized cost.

## Need help? Contact us

If you have questions or need help, [create a support request](#).

## Next steps

To learn more about Azure Reservations, see the following articles:

- [What are Azure Reservations?](#)
- [Prepay for Virtual Machines with Azure Reserved VM Instances](#)
- [Understand how the reservation discount is applied](#)
- [Windows software costs not included with Reservations](#)
- [Use service principal to get cost data](#)
- [Use cost management exports](#)

# Query

Reference

Service: Cost Management

API Version: 2023-03-01

## Operations

<a href="#">Usage</a>	Query the usage data for scope defined.
<a href="#">Usage By External Cloud Provider Type</a>	Query the usage data for external cloud provider type defined.

# Forecast

Reference

Service: Cost Management

API Version: 2023-03-01

## Operations

<a href="#">External Cloud Provider Usage</a>	Lists the forecast charges for external cloud provider type defined.
<a href="#">Usage</a>	Lists the forecast charges for scope defined.

# Dimensions

Reference

Service: Cost Management

API Version: 2023-03-01

## Operations

<a href="#">By External Cloud Provider Type</a>	Lists the dimensions by the external cloud provider type.
<a href="#">List</a>	Lists the dimensions by the defined scope.

# Views

Reference

Service: Cost Management

API Version: 2023-03-01

## Operations

<a href="#">Create Or Update</a>	The operation to create or update a view. Update operation requires latest eTag to be set in the request. You may obtain the latest eTag by performing a get ope...
<a href="#">Create Or Update By Scope</a>	The operation to create or update a view. Update operation requires latest eTag to be set in the request. You may obtain the latest eTag by performing a get ope...
<a href="#">Delete</a>	The operation to delete a view.
<a href="#">Delete By Scope</a>	The operation to delete a view.
<a href="#">Get</a>	Gets the view by view name.
<a href="#">Get By Scope</a>	Gets the view for the defined scope by view name.
<a href="#">List</a>	Lists all views by tenant and object.
<a href="#">List By Scope</a>	Lists all views at the given scope.

# Tutorial: Create and manage budgets

Article • 06/07/2023

Budgets in Cost Management help you plan for and drive organizational accountability. They help you proactively inform others about their spending to manage costs and monitor how spending progresses over time.

You can configure alerts based on your actual cost or forecasted cost to ensure that your spending is within your organizational spending limit. Notifications are triggered when the budget thresholds you've created are exceeded. Resources are not affected, and your consumption isn't stopped. You can use budgets to compare and track spending as you analyze costs.

Cost and usage data is typically available within 8-24 hours and budgets are evaluated against these costs every 24 hours. Be sure to get familiar with [Cost and usage data updates](#) specifics. When a budget threshold is met, email notifications are normally sent within an hour of the evaluation.

Budgets reset automatically at the end of a period (monthly, quarterly, or annually) for the same budget amount when you select an expiration date in the future. Because they reset with the same budget amount, you need to create separate budgets when budgeted currency amounts differ for future periods. When a budget expires, it's automatically deleted.

The examples in this tutorial walk you through creating and editing a budget for an Azure Enterprise Agreement (EA) subscription.

Watch the [Apply budgets to subscriptions using the Azure portal](#) video to see how you can create budgets in Azure to monitor spending. To watch other videos, visit the [Cost Management YouTube channel](#).

<https://www.youtube-nocookie.com/embed/UrkHiUx19Po>

In this tutorial, you learn how to:

- ✓ Create a budget in the Azure portal
- ✓ Create and edit budgets with PowerShell
- ✓ Create a budget with an Azure Resource Manager template

## Prerequisites

Budgets are supported for the following types of Azure account types and scopes:

- Azure role-based access control (Azure RBAC) scopes
  - Management groups
  - Subscription
- Enterprise Agreement scopes
  - Billing account
  - Department
  - Enrollment account
- Individual agreements
  - Billing account
- Microsoft Customer Agreement scopes
  - Billing account
  - Billing profile
  - Invoice section
  - Customer
- AWS scopes
  - External account
  - External subscription

To view budgets, you need at least read access for your Azure account.

If you have a new subscription, you can't immediately create a budget or use other Cost Management features. It might take up to 48 hours before you can use all Cost Management features.

For Azure EA subscriptions, you must have read access to view budgets. To create and manage budgets, you must have contributor permission.

The following Azure permissions, or scopes, are supported per subscription for budgets by user and group.

- Owner – Can create, modify, or delete budgets for a subscription.
- Contributor and Cost Management contributor – Can create, modify, or delete their own budgets. Can modify the budget amount for budgets created by others.
- Reader and Cost Management reader – Can view budgets that they have permission to.

For more information about scopes, including access needed to configure exports for Enterprise Agreement and Microsoft Customer agreement scopes, see [Understand and work with scopes](#). For more information about assigning permission to Cost Management data, see [Assign access to Cost Management data](#).

## Sign in to Azure

- Sign in to the Azure portal at <https://portal.azure.com>.

# Create a budget in the Azure portal

You can create an Azure subscription budget for a monthly, quarterly, or annual period.

To create or view a budget, open a scope in the Azure portal and select **Budgets** in the menu. For example, navigate to **Subscriptions**, select a subscription from the list, and then select **Budgets** in the menu. Use the **Scope** pill to switch to a different scope, like a management group, in Budgets. For more information about scopes, see [Understand and work with scopes](#).

After you create budgets, they show a simple view of your current spending against them.

Select **Add**.

Name	Scope	Reset period	Creation date	Expiration date	Budget	Forecasted	Evaluated spend	Progress
EAAccount-BoTest...	8608480 (Billing acc...)	Monthly	2/1/2021	1/31/2023	\$100.00	\$0.00	\$0.00	0.00%
EAAccount-BoTest-A...	208903 (Enrollment ...)	Monthly	2/1/2021	1/31/2023	\$100.00	\$0.00	\$0.00	0.00%
Pri_Forecast	8608480 (Billing acc...)	Monthly	2/1/2021	2/28/2022	\$100.00	\$120.5K	\$26,046	100.00%
Pri_actualForecast	8608480 (Billing acc...)	Monthly	2/1/2021	1/31/2023	\$200.00	\$120.5K	\$26,046	100.00%
EAAccount-BoTest...	84820 (Department)	Monthly	2/1/2021	1/31/2023	\$100.00	\$4,572	\$1,088	100.00%
Pri_EdgecaseTest	8608480 (Billing acc...)	Monthly	2/1/2021	2/28/2022	\$100.00	\$515.23	\$131.43	100.00%
ACM_Department_B...	84820 (Department)	Monthly	10/1/2019	9/30/2021	\$55,000.00	○	\$4,272	7.71%
Enrollment_budget	8608480 (Billing acc...)	Monthly	4/1/2020	3/31/2022	\$45,000.00	○	\$26,046	57.88%
ACM	8608480 (Billing acc...)	Monthly	8/1/2020	7/31/2022	\$30,000.00	○	\$0.00	0.00%
JoTestBudget	8608480 (Billing acc...)	Monthly	11/1/2020	10/31/2022	\$50,000.00	○	\$26,046	52.09%
DemoTestBudget	8608480 (Billing acc...)	Monthly	12/1/2020	11/30/2022	\$40,000.00	○	\$26,046	65.12%
ClaroITGT	8608480 (Billing acc...)	Monthly	12/1/2020	11/30/2022	\$35,000.00	○	\$26,046	74.42%

In the **Create budget** window, make sure that the scope shown is correct. Choose any filters that you want to add. Filters allow you to create budgets on specific costs, such as resource groups in a subscription or a service like virtual machines. For more information about the common filter properties that you can use in budgets and cost analysis, see [Group and filter properties](#).

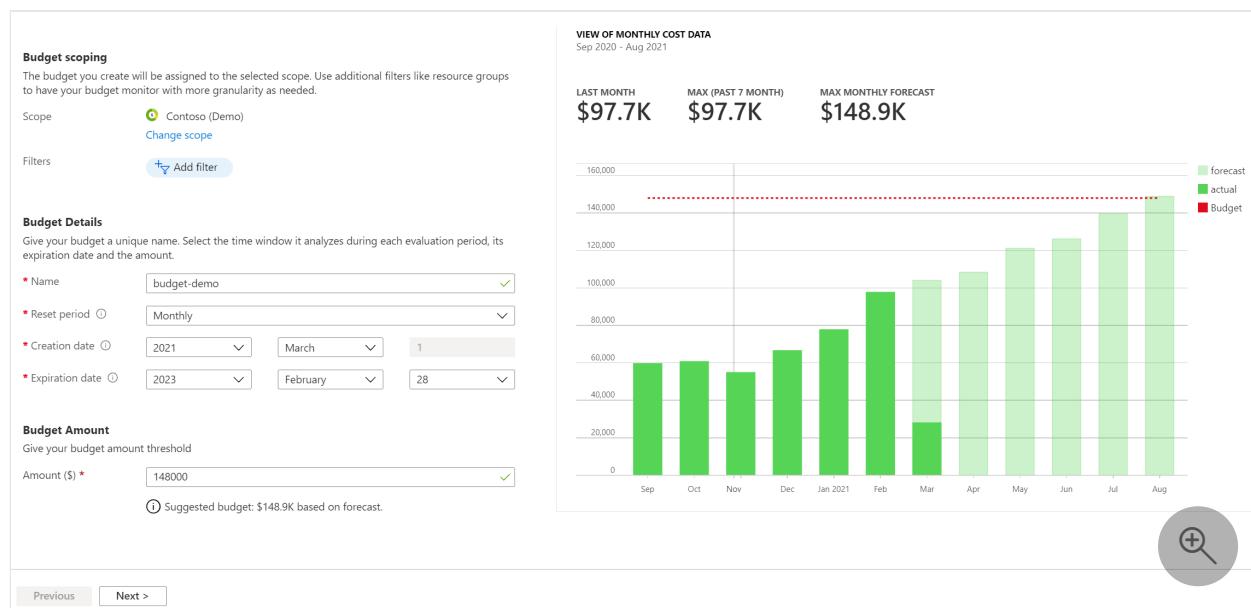
After you identify your scope and filters, type a budget name. Then, choose a monthly, quarterly, or annual budget reset period. The reset period determines the time window that's analyzed by the budget. The cost evaluated by the budget starts at zero at the beginning of each new period. When you create a quarterly budget, it works in the same

way as a monthly budget. The difference is that the budget amount for the quarter is evenly divided among the three months of the quarter. An annual budget amount is evenly divided among all 12 months of the calendar year.

If you have a Pay-As-You-Go, MSDN, or Visual Studio subscription, your invoice billing period might not align to the calendar month. For those subscription types and resource groups, you can create a budget that's aligned to your invoice period or to calendar months. To create a budget aligned to your invoice period, select a reset period of **Billing month**, **Billing quarter**, or **Billing year**. To create a budget aligned to the calendar month, select a reset period of **Monthly**, **Quarterly**, or **Annually**.

Next, identify the expiration date when the budget becomes invalid and stops evaluating your costs.

Based on the fields chosen in the budget so far, a graph is shown to help you select a threshold to use for your budget. The suggested budget is based on the highest forecasted cost that you might incur in future periods. You can change the budget amount.



After you configure the budget amount, select **Next** to configure budget alerts for actual cost and forecasted budget alerts.

## Configure actual costs budget alerts

Budgets require at least one cost threshold (% of budget) and a corresponding email address. You can optionally include up to five thresholds and five email addresses in a single budget. When a budget threshold is met, email notifications are normally sent within an hour of the evaluation. Actual costs budget alerts are generated for the actual cost you've accrued in relation to the budget thresholds configured.

# Configure forecasted budget alerts

Forecasted alerts provide advanced notification that your spending trends are likely to exceed your budget. The alerts use forecasted cost predictions. Alerts are generated when the forecasted cost projection exceeds the set threshold. You can configure a forecasted threshold (% of budget). When a forecasted budget threshold is met, notifications are normally sent within an hour of the evaluation.

To toggle between configuring an Actual vs Forecasted cost alert, use the `Type` field when configuring the alert as shown in the following image.

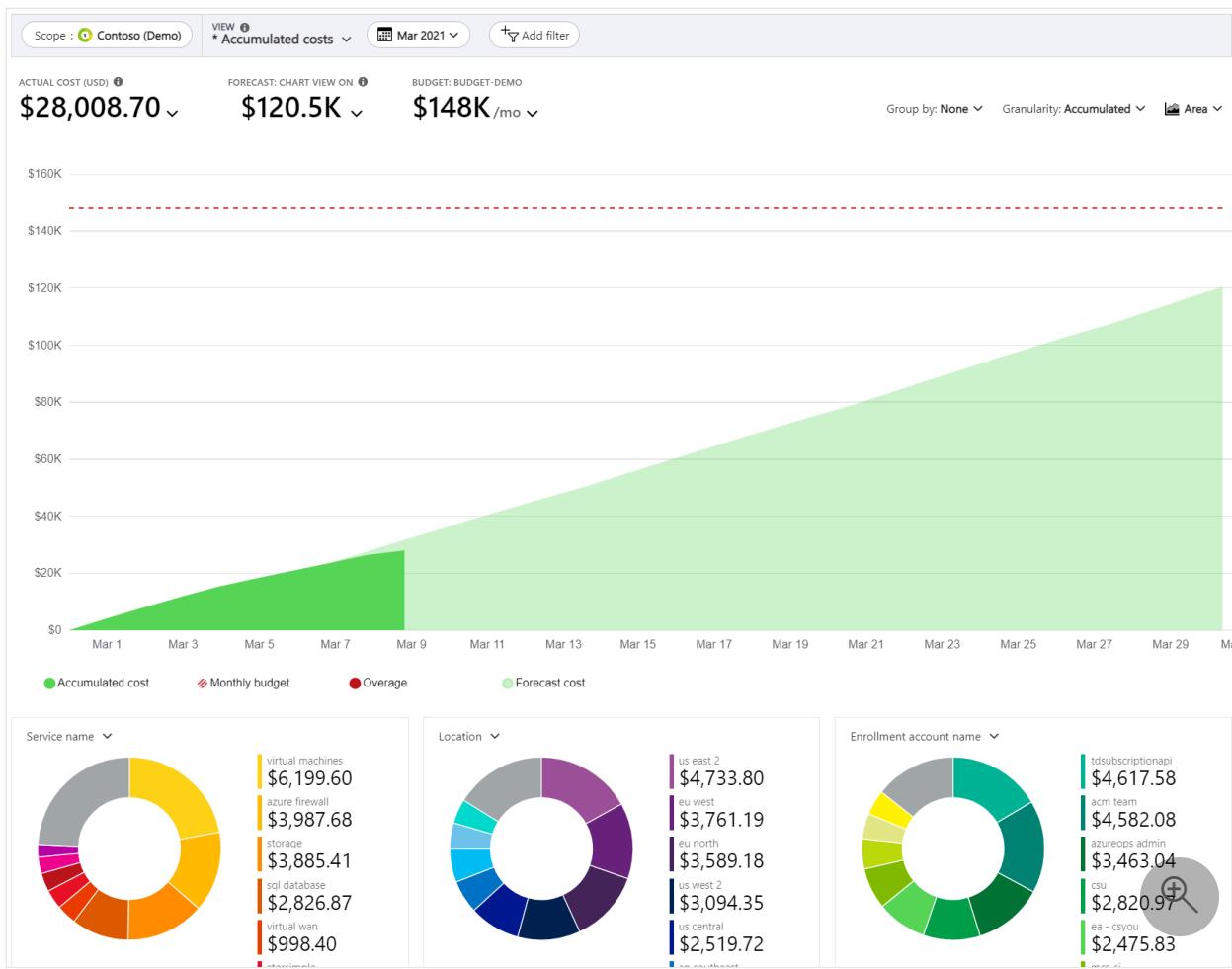
If you want to receive emails, add `azure-noreply@microsoft.com` to your approved senders list so that emails don't go to your junk email folder. For more information about notifications, see [Use cost alerts](#).

In the following example, an email alert gets generated when 90% of the budget is reached. If you create a budget with the Budgets API, you can also assign roles to people to receive alerts. Assigning roles to people isn't supported in the Azure portal. For more about the Budgets API, see [Budgets API](#). If you want to have an email alert sent in a different language, see [Supported locales for budget alert emails](#).

Alert limits support a range of 0.01% to 1000% of the budget threshold that you've provided.

The screenshot shows the 'Create a budget' page in the Azure portal. At the top, there are two tabs: 'Create a budget' (selected) and 'Set alerts'. Below the tabs, a note says 'Configure alert conditions and send email notifications based on your spend.' Under the 'Alert conditions' section, there are two rows for 'Type', '% of budget', and 'Amount'. The first row has 'Actual' selected with 90% and \$133200. The second row has 'Forecasted' selected with 100% and \$148000. There is also a 'Select type' dropdown and an 'Enter %' input field. In the 'Alert recipients (email)' section, there are two entries: 'user@contoso.com' and 'example@email.com'. A note below says 'It is recommended to add azure-noreply@microsoft.com to your email white list to ensure alert mails do not go to your spam folder.' Under 'Language preference', it says 'Select your preferred language for receiving the alert email for all recipients provided above. Default is the language associated to your enrollment.' A dropdown menu shows 'Languages \*' and 'Default'. A note at the bottom left says 'Your budget evaluation will begin in a few hours. Learn more'. At the bottom right is a circular button with a magnifying glass icon. At the very bottom are 'Previous' and 'Create' buttons.

After you create a budget, it's shown in cost analysis. Viewing your budget against your spending trend is one of the first steps when you start to [analyze your costs and spending](#).



In the preceding example, you created a budget for a subscription. You can also create a budget for a resource group. If you want to create a budget for a resource group, navigate to **Cost Management + Billing > Subscriptions >** select a subscription **> Resource groups >** select a resource group **> Budgets >** and then **Add a budget**.

## Create a budget for combined Azure and AWS costs

You can group your Azure and AWS costs together by assigning a management group to your connector along with its consolidated and linked accounts. Assign your Azure subscriptions to the same management group. Then create a budget for the combined costs.

1. In Cost Management, select **Budgets**.
2. Select **Add**.
3. Select **Change scope** and then select the management group.
4. Continue creating the budget until complete.

## Costs in budget evaluations

Budget cost evaluations now include reserved instance and purchase data. If the charges apply to you, then you might receive alerts as charges are incorporated into your

evaluations. Sign in to the [Azure portal](#) to verify that budget thresholds are properly configured to account for the new costs. Your Azure billed charges aren't changed. Budgets now evaluate against a more complete set of your costs. If the charges don't apply to you, then your budget behavior remains unchanged.

If you want to filter the new costs so that budgets are evaluated against first party Azure consumption charges only, add the following filters to your budget:

- Publisher Type: Azure
- Charge Type: Usage

Budget cost evaluations are based on actual cost. They don't include amortization. For more information about filtering options available to you in budgets, see [Understanding grouping and filtering options](#).

## Trigger an action group

When you create or edit a budget for a subscription or resource group scope, you can configure it to call an action group. The action group can perform various actions when your budget threshold is met. You can receive mobile push notifications when your budget threshold is met by enabling [Azure app push notifications](#) while configuring the action group.

Action groups are currently only supported for subscription and resource group scopes. For more information about creating action groups, see [action groups](#).

For more information about using budget-based automation with action groups, see [Manage costs with budgets](#).

To create or update action groups, select **Manage action group** while you're creating or editing a budget.

Create a budget

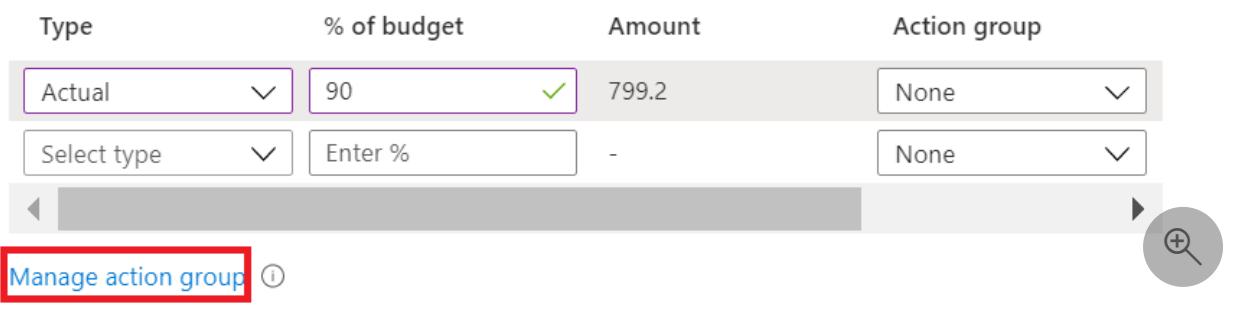
**2 Set alerts**

Configure alert conditions and send email notifications based on your spend.

**\* Alert conditions**

Type	% of budget	Amount	Action group
Actual	90	799.2	None
Select type	Enter %	-	None

**Manage action group** ⓘ



Next, select **Add action group** and create the action group.

Budget integration with action groups works for action groups that have enabled or disabled common alert schema. For more information on how to enable common alert schema, see [How do I enable the common alert schema?](#)

## Budgets in the Azure mobile app

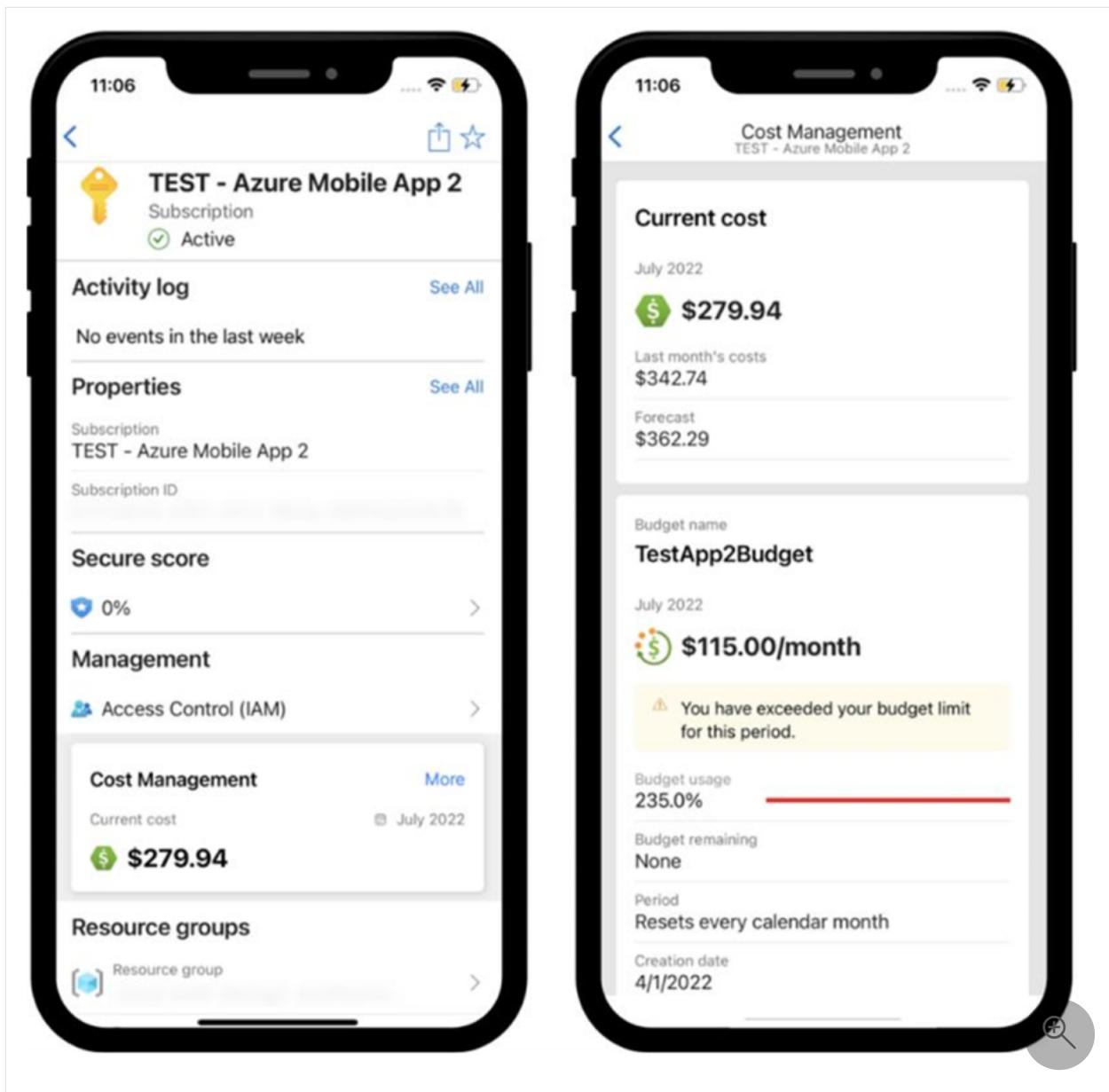
You can view budgets for your subscriptions and resource groups from the **Cost Management** card in the [Azure app](#).

1. Navigate to any subscription or resource group.
2. Find the **Cost Management** card and tap **More**.
3. Budgets load below the **Current cost** card. They're sorted by descending order of usage.

To receive mobile push notifications when your budget threshold is met, you can configure action groups. When setting up budget alerts, make sure to select an action group that has [Azure app push notifications](#) enabled.

**ⓘ Note**

Currently, the Azure mobile app only supports the subscription and resource group scopes for budgets.



## Create and edit budgets with PowerShell

If you're an EA customer, you can create and edit budgets programmatically using the Azure PowerShell module. However, we recommend that you use REST APIs to create and edit budgets because CLI commands might not support the latest version of the APIs.

### ⓘ Note

Customers with a Microsoft Customer Agreement should use the [Budgets REST API](#) to create budgets programmatically.

To download the latest version of Azure PowerShell, run the following command:

```
Azure PowerShell
```

```
install-module -name Az
```

The following example commands create a budget.

Azure PowerShell

```
#Sign into Azure PowerShell with your account

Connect-AzAccount

#Select a subscription to monitor with a budget

select-AzSubscription -Subscription "Your Subscription"

#Create an action group email receiver and corresponding action group

$email1 = New-AzActionGroupReceiver -EmailAddress test@test.com -Name EmailReceiver1
$ActionGroupId = (Set-AzActionGroup -ResourceGroupName YourResourceGroup -
Name TestAG -ShortName TestAG -Receiver $email1).Id

#Create a monthly budget that sends an email and triggers an Action Group to
send a second email. Make sure the StartDate for your monthly budget is set
to the first day of the current month. Note that Action Groups can also be
used to trigger automation such as Azure Functions or Webhooks.

Get-AzContext
New-AzConsumptionBudget -Amount 100 -Name TestPSBudget -Category Cost -
StartDate 2020-02-01 -TimeGrain Monthly -EndDate 2022-12-31 -ContactEmail
test@test.com -NotificationKey Key1 -NotificationThreshold 0.8 -
NotificationEnabled -ContactGroup $ActionGroupId
```

## Create a budget with an Azure Resource Manager template

You can create a budget using an Azure Resource Manager template. To use the template, see [Create a budget with an Azure Resource Manager template](#).

## Clean up resources

If you created a budget and you no longer need it, view its details and delete it.

## Next steps

In this tutorial, you learned how to:

- ✓ Create a budget in the Azure portal
- ✓ Create and edit budgets with PowerShell
- ✓ Create a budget with an Azure Resource Manager template

Advance to the next tutorial to create a recurring export for your cost management data.

[Create and manage exported data](#)

# Quickstart: Create a budget with Bicep

Article • 03/08/2023

Budgets in Cost Management help you plan for and drive organizational accountability. With budgets, you can account for the Azure services you consume or subscribe to during a specific period. They help you inform others about their spending to proactively manage costs and monitor how spending progresses over time. When the budget thresholds you've created are exceeded, notifications are triggered. None of your resources are affected and your consumption isn't stopped. You can use budgets to compare and track spending as you analyze costs. This quickstart shows you how to create a budget named 'MyBudget' using Bicep.

[Bicep](#) is a domain-specific language (DSL) that uses declarative syntax to deploy Azure resources. It provides concise syntax, reliable type safety, and support for code reuse. Bicep offers the best authoring experience for your infrastructure-as-code solutions in Azure.

## Prerequisites

If you don't have an Azure subscription, create a [free account](#) before you begin.

If you have a new subscription, you can't immediately create a budget or use other Cost Management features. It might take up to 48 hours before you can use all Cost Management features.

Budgets are supported for the following types of Azure account types and scopes:

- Azure role-based access control (Azure RBAC) scopes
  - Management groups
  - Subscription
- Enterprise Agreement scopes
  - Billing account
  - Department
  - Enrollment account
- Individual agreements
  - Billing account
- Microsoft Customer Agreement scopes
  - Billing account
  - Billing profile
  - Invoice section
  - Customer

- AWS scopes
  - External account
  - External subscription

To view budgets, you need at least read access for your Azure account.

For Azure EA subscriptions, you must have read access to view budgets. To create and manage budgets, you must have contributor permission.

The following Azure permissions, or scopes, are supported per subscription for budgets by user and group. For more information about scopes, see [Understand and work with scopes](#).

- Owner: Can create, modify, or delete budgets for a subscription.
- Contributor and Cost Management contributor: Can create, modify, or delete their own budgets. Can modify the budget amount for budgets created by others.
- Reader and Cost Management reader: Can view budgets that they have permission to.

For more information about assigning permission to Cost Management data, see [Assign access to Cost Management data](#).

## No filter

### Review the Bicep file

The Bicep file used in this quickstart is from [Azure Quickstart Templates](#).

```
Bicep

targetScope = 'subscription'

@description('Name of the Budget. It should be unique within a resource
group.')
param budgetName string = 'MyBudget'

@description('The total amount of cost or usage to track with the budget')
param amount int = 1000

@description('The time covered by a budget. Tracking of the amount will be
reset based on the time grain.')
@allowed([
 'Monthly'
 'Quarterly'
 'Annually'
])
])
```

```

param timeGrain string = 'Monthly'

@description('The start date must be first of the month in YYYY-MM-DD
format. Future start date should not be more than three months. Past start
date should be selected within the timegrain preiod.')
param startDate string

@description('The end date for the budget in YYYY-MM-DD format. If not
provided, we default this to 10 years from the start date.')
param endDate string

@description('Threshold value associated with a notification. Notification
is sent when the cost exceeded the threshold. It is always percent and has
to be between 0.01 and 1000.')
param firstThreshold int = 90

@description('Threshold value associated with a notification. Notification
is sent when the cost exceeded the threshold. It is always percent and has
to be between 0.01 and 1000.')
param secondThreshold int = 110

@description('The list of email addresses to send the budget notification to
when the threshold is exceeded.')
param contactEmails array

resource budget 'Microsoft.Consumption/budgets@2021-10-01' = {
 name: budgetName
 properties: {
 timePeriod: {
 startDate: startDate
 endDate: endDate
 }
 timeGrain: timeGrain
 amount: amount
 category: 'Cost'
 notifications: {
 NotificationForExceededBudget1: {
 enabled: true
 operator: 'GreaterThan'
 threshold: firstThreshold
 contactEmails: contactEmails
 }
 NotificationForExceededBudget2: {
 enabled: true
 operator: 'GreaterThan'
 threshold: secondThreshold
 contactEmails: contactEmails
 }
 }
 }
}

```

One Azure resource is defined in the Bicep file:

- Microsoft.Consumption/budgets: Create a budget.

## Deploy the Bicep file

1. Save the Bicep file as `main.bicep` to your local computer.
2. Deploy the Bicep file using either Azure CLI or Azure PowerShell.

CLI

```
Azure CLI

myContactEmails =('user1@contoso.com", "user2@contoso.com")

az deployment sub create --name demoSubDeployment --location
centralus --template-file main.bicep --parameters startDate=<start-
date> endDate=<end-date> contactEmails=$myContactEmails
```

You need to enter the following parameters:

- **startDate**: Replace `<start-date>` with the start date. It must be the first of the month in YYYY-MM-DD format. A future start date shouldn't be more than three months in the future. A past start date should be selected within the timegrain period.
- **endDate**: Replace `<end-date>` with the end date in YYYY-MM-DD format. If not provided, it defaults to ten years from the start date.
- **contactEmails**: First create a variable that holds your emails and then pass that variable. Replace the sample emails with the email addresses to send the budget notification to when the threshold is exceeded.

 **Note**

When the deployment finishes, you should see a message indicating the deployment succeeded.

## One filter

### Review the Bicep file

The Bicep file used in this quickstart is from [Azure Quickstart Templates](#).

## Bicep

```
targetScope = 'subscription'

@description('Name of the Budget. It should be unique within a resource group.')
param budgetName string = 'MyBudget'

@description('The total amount of cost or usage to track with the budget')
param amount int = 1000

@description('The time covered by a budget. Tracking of the amount will be reset based on the time grain.')
@allowed([
 'Monthly'
 'Quarterly'
 'Annually'
])
param timeGrain string = 'Monthly'

@description('The start date must be first of the month in YYYY-MM-DD format. Future start date should not be more than three months. Past start date should be selected within the timegrain preiod.')
param startDate string

@description('The end date for the budget in YYYY-MM-DD format. If not provided, we default this to 10 years from the start date.')
param endDate string

@description('Threshold value associated with a notification. Notification is sent when the cost exceeded the threshold. It is always percent and has to be between 0.01 and 1000.')
param firstThreshold int = 90

@description('Threshold value associated with a notification. Notification is sent when the cost exceeded the threshold. It is always percent and has to be between 0.01 and 1000.')
param secondThreshold int = 110

@description('The list of email addresses to send the budget notification to when the threshold is exceeded.')
param contactEmails array

@description('The set of values for the resource group filter.')
param resourceGroupFilterValues array

resource budget 'Microsoft.Consumption/budgets@2021-10-01' = {
 name: budgetName
 properties: {
 timePeriod: {
 startDate: startDate
 endDate: endDate
 }
 timeGrain: timeGrain
 }
}
```

```

amount: amount
category: 'Cost'
notifications: {
 NotificationForExceededBudget1: {
 enabled: true
 operator: 'GreaterThan'
 threshold: firstThreshold
 contactEmails: contactEmails
 }
 NotificationForExceededBudget2: {
 enabled: true
 operator: 'GreaterThan'
 threshold: secondThreshold
 contactEmails: contactEmails
 }
}
filter: {
 dimensions: {
 name: 'ResourceGroupName'
 operator: 'In'
 values: resourceGroupFilterValues
 }
}
}
}

```

One Azure resource is defined in the Bicep file:

- [Microsoft.Consumption/budgets](#): Create a budget.

## Deploy the Bicep file

1. Save the Bicep file as **main.bicep** to your local computer.
2. Deploy the Bicep file using either Azure CLI or Azure PowerShell.

CLI

Azure CLI

```

myContactEmails =('user1@contoso.com", "user2@contoso.com")'
myRgFilterValues =("resource-group-01", "resource-group-02")

az deployment sub create --name demoSubDeployment --location
centralus --template-file main.bicep --parameters startDate=<start-
date> endDate=<end-date> contactEmails=$myContactEmails
resourceGroupFilterValues=$myRgFilterValues

```

You need to enter the following parameters:

- **startDate**: Replace <start-date> with the start date. It must be the first of the month in YYYY-MM-DD format. A future start date shouldn't be more than three months in the future. A past start date should be selected within the timegrain period.
- **endDate**: Replace <end-date> with the end date in YYYY-MM-DD format. If not provided, it defaults to ten years from the start date.
- **contactEmails**: First create a variable that holds your emails and then pass that variable. Replace the sample emails with the email addresses to send the budget notification to when the threshold is exceeded.
- **resourceGroupFilterValues**: First create a variable that holds your resource group filter values and then pass that variable. Replace the sample filter values with the set of values for your resource group filter.

① Note

When the deployment finishes, you should see a message indicating the deployment succeeded.

## Two or more filters

### Review the Bicep file

The Bicep file used in this quickstart is from [Azure Quickstart Templates](#).

Bicep

```
targetScope = 'subscription'

@description('Name of the Budget. It should be unique within a resource
group.')
param budgetName string = 'MyBudget'

@description('The total amount of cost or usage to track with the budget')
param amount int = 1000

@description('The time covered by a budget. Tracking of the amount will be
reset based on the time grain.')
@allowed([
 'Monthly'
 'Quarterly'
 'Annually'
])
```

```

param timeGrain string = 'Monthly'

@description('The start date must be first of the month in YYYY-MM-DD
format. Future start date should not be more than three months. Past start
date should be selected within the timegrain preiod.')
param startDate string

@description('The end date for the budget in YYYY-MM-DD format. If not
provided, we default this to 10 years from the start date.')
param endDate string

@description('Threshold value associated with a notification. Notification
is sent when the cost exceeded the threshold. It is always percent and has
to be between 0.01 and 1000.')
param firstThreshold int = 90

@description('Threshold value associated with a notification. Notification
is sent when the cost exceeded the threshold. It is always percent and has
to be between 0.01 and 1000.')
param secondThreshold int = 110

@description('The list of contact roles to send the budget notification to
when the threshold is exceeded.')
param contactRoles array =
 [
 'Owner'
 'Contributor'
 'Reader'
]

@description('The list of email addresses to send the budget notification to
when the threshold is exceeded.')
param contactEmails array

@description('The list of action groups to send the budget notification to
when the threshold is exceeded. It accepts array of strings.')
param contactGroups array

@description('The set of values for the resource group filter.')
param resourceGroupFilterValues array

@description('The set of values for the meter category filter.')
param meterCategoryFilterValues array

resource budget 'Microsoft.Consumption/budgets@2021-10-01' = {
 name: budgetName
 properties: {
 timePeriod: {
 startDate: startDate
 endDate: endDate
 }
 timeGrain: timeGrain
 amount: amount
 category: 'Cost'
 notifications: {
 NotificationForExceededBudget1: {

```

```

 enabled: true
 operator: 'GreaterThan'
 threshold: firstThreshold
 contactEmails: contactEmails
 contactRoles: contactRoles
 contactGroups: contactGroups
 }
 NotificationForExceededBudget2: {
 enabled: true
 operator: 'GreaterThan'
 threshold: secondThreshold
 contactEmails: contactEmails
 contactRoles: contactRoles
 contactGroups: contactGroups
 thresholdType: 'Forecasted'
 }
}
filter: {
 and: [
 {
 dimensions: {
 name: 'ResourceGroupName'
 operator: 'In'
 values: resourceGroupFilterValues
 }
 }
 {
 dimensions: {
 name: 'MeterCategory'
 operator: 'In'
 values: meterCategoryFilterValues
 }
 }
]
}
}
}

```

One Azure resource is defined in the Bicep file:

- [Microsoft.Consumption/budgets](#): Create a budget.

## Deploy the Bicep file

1. Save the Bicep file as **main.bicep** to your local computer.
2. Deploy the Bicep file using either Azure CLI or Azure PowerShell.

CLI

## Azure CLI

```
myContactEmails =('user1@contoso.com", "user2@contoso.com")'
myContactGroups =('/subscriptions/{sub-id}/resourceGroups/{rg-
name}/providers/microsoft.insights/actionGroups/groupone',
'/subscriptions/{sub-id}/resourceGroups/{rg-
name}/providers/microsoft.insights/actionGroups/grouptwo')'
myRgFilterValues =("resource-group-01", "resource-group-02")'
myMeterCategoryFilterValues =("meter-category-01", "meter-
category-02")'

az deployment sub create --name demoSubDeployment --location
centralus --template-file main.bicep --parameters startDate=<start-
date> endDate=<end-date> contactEmails=$myContactEmails
contactGroups=$myContactGroups
resourceGroupFilterValues=$myRgFilterValues
meterCategoryFilterValues=$myMeterCategoryFilterValues
```

You need to enter the following parameters:

- **startDate**: Replace <start-date> with the start date. It must be the first of the month in YYYY-MM-DD format. A future start date shouldn't be more than three months in the future. A past start date should be selected within the timegrain period.
- **endDate**: Replace <end-date> with the end date in YYYY-MM-DD format. If not provided, it defaults to ten years from the start date.
- **contactEmails**: First create a variable that holds your emails and then pass that variable. Replace the sample emails with the email addresses to send the budget notification to when the threshold is exceeded.
- **contactGroups**: First create a variable that holds your contact groups and then pass that variable. Replace the sample contact groups with the list of action groups to send the budget notification to when the threshold is exceeded. You must pass the resource ID of the action group, which you can get with `az monitor action-group show` or `Get-AzActionGroup`.
- **resourceGroupFilterValues**: First create a variable that holds your resource group filter values and then pass that variable. Replace the sample filter values with the set of values for your resource group filter.
- **meterCategoryFilterValues**: First create a variable that holds your meter category filter values and then pass that variable. Replace the sample filter values within parentheses with the set of values for your meter category filter.

### (!) Note

When the deployment finishes, you should see a message indicating the deployment succeeded.

## Review deployed resources

Use the Azure portal, Azure CLI, or Azure PowerShell to list the deployed resources in the resource group.

CLI

Azure CLI

```
az consumption budget list
```

## Clean up resources

When you no longer need the budget, use the Azure portal, Azure CLI, or Azure PowerShell to delete it:

CLI

Azure CLI

```
az consumption budget delete --budget-name MyBudget
```

## Next steps

In this quickstart, you created a budget and deployed it using Bicep. To learn more about Cost Management and Billing and Bicep, continue on to the articles below.

- Read the [Cost Management and Billing](#) overview.
- [Create budgets](#) in the Azure portal.
- Learn more about [Bicep](#).

# Quickstart: Create a budget with an ARM template

Article • 04/05/2023

Budgets in Cost Management help you plan for and drive organizational accountability. With budgets, you can account for the Azure services you consume or subscribe to during a specific period. They help you inform others about their spending to proactively manage costs, and to monitor how spending progresses over time. When the budget thresholds you've created are exceeded, notifications are triggered. None of your resources are affected and your consumption isn't stopped. You can use budgets to compare and track spending as you analyze costs. This quickstart shows you how to create a budget using three different Azure Resource Manager templates (ARM template).

A [resource manager template](#) is a JavaScript Object Notation (JSON) file that defines the infrastructure and configuration for your project. The template uses declarative syntax. In declarative syntax, you describe your intended deployment without writing the sequence of programming commands to create the deployment.

If your environment meets the prerequisites and you're familiar with using ARM templates, select the **Deploy to Azure** button for one of the following templates. The template will open in the Azure portal.

Template	Deployment button
No filter	 Deploy to Azure
One filter	 Deploy to Azure
Two or more filters	 Deploy to Azure

## Prerequisites

If you don't have an Azure subscription, create a [free account](#) before you begin.

If you have a new subscription, you can't immediately create a budget or use other Cost Management features. It might take up to 48 hours before you can use all Cost Management features.

Budgets are supported for the following types of Azure account types and scopes:

- Azure role-based access control (Azure RBAC) scopes
  - Management groups
  - Subscription
- Enterprise Agreement scopes
  - Billing account
  - Department
  - Enrollment account
- Individual agreements
  - Billing account
- Microsoft Customer Agreement scopes
  - Billing account
  - Billing profile
  - Invoice section
  - Customer
- AWS scopes
  - External account
  - External subscription

To view budgets, you need at least read access for your Azure account.

For Azure EA subscriptions, you must have read access to view budgets. To create and manage budgets, you must have contributor permission.

The following Azure permissions, or scopes, are supported per subscription for budgets by user and group. For more information about scopes, see [Understand and work with scopes](#).

- Owner – Can create, modify, or delete budgets for a subscription.
- Contributor and Cost Management contributor – Can create, modify, or delete their own budgets. Can modify the budget amount for budgets created by others.
- Reader and Cost Management reader – Can view budgets that they have permission to.

For more information about assigning permission to Cost Management data, see [Assign access to Cost Management data](#).

Use one of the following templates, based on your needs.

Template	Description
No filter	The ARM template doesn't have any filters.

Template	Description
One filter	The ARM template has a filter for resource groups.
Two or more filters	The ARM template has a filter for resource groups and a filter for meter categories.

## Review and deploy the template

No filter

### Review the template

The template used in this quickstart is from [Azure Quickstart Templates](#).

JSON

```
{
 "$schema": "https://schema.management.azure.com/schemas/2018-05-01/subscriptionDeploymentTemplate.json#",
 "contentVersion": "1.0.0.0",
 "metadata": {
 "_generator": {
 "name": "bicep",
 "version": "0.5.6.12127",
 "templateHash": "15851432186142062555"
 }
 },
 "parameters": {
 "budgetName": {
 "type": "string",
 "defaultValue": "MyBudget",
 "metadata": {
 "description": "Name of the Budget. It should be unique within a resource group."
 }
 },
 "amount": {
 "type": "int",
 "defaultValue": 1000,
 "metadata": {
 "description": "The total amount of cost or usage to track with the budget"
 }
 },
 "timeGrain": {
 "type": "string",
 "defaultValue": "Monthly",
 "metadata": {
 "description": "The time grain for the budget"
 }
 }
 }
}
```

```
 "allowedValues": [
 "Monthly",
 "Quarterly",
 "Annually"
],
 "metadata": {
 "description": "The time covered by a budget. Tracking of the amount will be reset based on the time grain."
 }
},
"startDate": {
 "type": "string",
 "metadata": {
 "description": "The start date must be first of the month in YYYY-MM-DD format. Future start date should not be more than three months. Past start date should be selected within the timegrain preiod."
 }
},
"endDate": {
 "type": "string",
 "metadata": {
 "description": "The end date for the budget in YYYY-MM-DD format. If not provided, we default this to 10 years from the start date."
 }
},
"firstThreshold": {
 "type": "int",
 "defaultValue": 90,
 "metadata": {
 "description": "Threshold value associated with a notification. Notification is sent when the cost exceeded the threshold. It is always percent and has to be between 0.01 and 1000."
 }
},
"secondThreshold": {
 "type": "int",
 "defaultValue": 110,
 "metadata": {
 "description": "Threshold value associated with a notification. Notification is sent when the cost exceeded the threshold. It is always percent and has to be between 0.01 and 1000."
 }
},
"contactEmails": {
 "type": "array",
 "metadata": {
 "description": "The list of email addresses to send the budget notification to when the threshold is exceeded."
 }
},
"resources": [
{
 "type": "Microsoft.Consumption/budgets",
```

```

 "apiVersion": "2021-10-01",
 "name": "[parameters('budgetName')]",
 "properties": {
 "timePeriod": {
 "startDate": "[parameters('startDate')]",
 "endDate": "[parameters('endDate')]"
 },
 "timeGrain": "[parameters('timeGrain')]",
 "amount": "[parameters('amount')]",
 "category": "Cost",
 "notifications": {
 "NotificationForExceededBudget1": {
 "enabled": true,
 "operator": "GreaterThan",
 "threshold": "[parameters('firstThreshold')]",
 "contactEmails": "[parameters('contactEmails')]"
 },
 "NotificationForExceededBudget2": {
 "enabled": true,
 "operator": "GreaterThan",
 "threshold": "[parameters('secondThreshold')]",
 "contactEmails": "[parameters('contactEmails')]"
 }
 }
 }
}
]
}

```

One Azure resource is defined in the template:

- [Microsoft.Consumption/budgets](#): Create a budget.

## Deploy the template

1. Select the following image to sign in to Azure and open a template. The template creates a budget without any filters.



2. Select or enter the following values.

## Create a Budget

...

Azure quickstart template

[Basics](#)   [Review + create](#)**Template**[create-budget-simple ↗](#)

1 resource

[Edit template](#)[Edit parameters](#)[Visualize](#)**Project details**

Deploying templates at subscription scope enables scenarios like applying policies and assigning roles at the subscription level. Subscription scope deployments are also used for creating resource groups and deploying resources in it. You can change the deployment scope by updating the schema in the template.

Subscription \* ⓘ

Trey Research R&amp;D Playground

**Instance details**

Region \* ⓘ

East US 2



Budget Name ⓘ

MyBudget

Amount ⓘ

1000

Time Grain ⓘ

Monthly



Start Date \* ⓘ

2021-01-01



End Date \* ⓘ

2021-03-01



First Threshold ⓘ

90

Second Threshold ⓘ

110

Contact Emails \* ⓘ

["user@contoso.com"]

[Review + create](#)

&lt; Previous

Next : Review + create &gt;

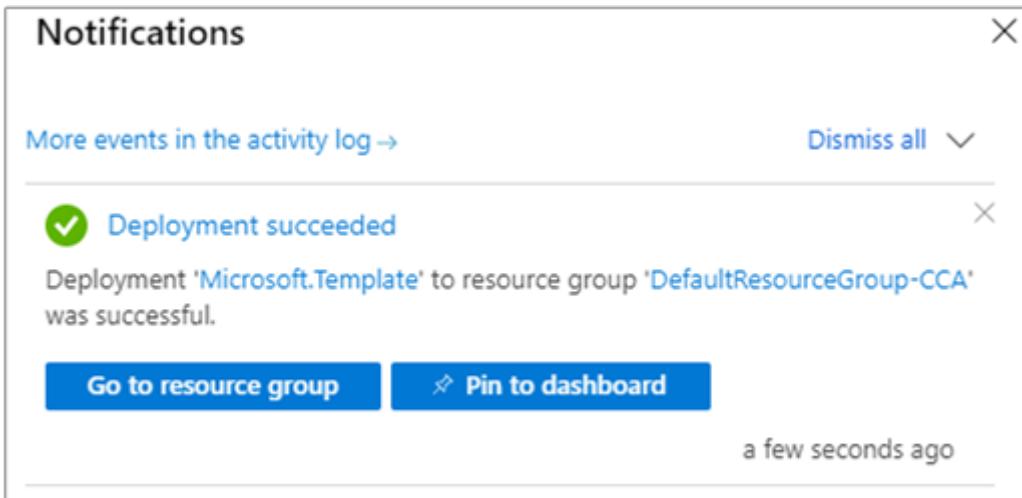
- **Subscription:** select an Azure subscription.
- **Resource group:** if required, select an existing resource group, or [Create new](#).
- **Region:** select an Azure region. For example, **Central US**.
- **Budget Name:** enter a name for the budget. It should be unique within a resource group. Only alphanumeric, underscore, and hyphen characters are allowed.
- **Amount:** enter the total amount of cost to track with the budget.
- **Time Grain:** enter the time covered by a budget. Allowed values are Monthly, Quarterly, or Annually. The budget resets at the end of the time grain.
- **Start Date:** enter the start date with the first day of the month in YYYY-MM-DD format. A future start date shouldn't be more than three months from today. You can specify a past start date with the Time Grain period.

- **End Date:** enter the end date for the budget in YYYY-MM-DD format.
- **First Threshold:** enter a threshold value for the first notification. A notification is sent when the cost exceeds the threshold. It's always percent and has to be between 0.01 and 1000.
- **Second Threshold:** enter a threshold value for the second notification. A notification is sent when the cost exceeds the threshold. It's always percent and has to be between 0.01 and 1000.
- **Contact Emails** enter a list of email addresses to send the budget notification to when a threshold is exceeded. It accepts an array of strings. Expected format is `["user1@domain.com", "user2@domain.com"]`.

3. Depending on your Azure subscription type, do one of the following actions:

- Select **Review + create**.
- Review the terms and conditions, select **I agree to the terms and conditions stated above**, and then select **Purchase**.

4. If you selected **Review + create**, your template is validated. Select **Create**.



The Azure portal is used to deploy the template. In addition to the Azure portal, you can also use Azure PowerShell, Azure CLI, and REST API. To learn about other deployment templates, see [Deploy templates](#).

## Validate the deployment

Use one of the following ways to verify that the budget is created.

Azure portal

Navigate to **Cost Management + Billing** > select a scope > **Budgets**.

# Clean up resources

When you no longer need a budget, delete it by using one the following methods:

## Azure portal

Navigate to **Cost Management + Billing** > select a billing scope > **Budgets** > select a budget > then select **Delete budget**.

## Next steps

In this quickstart, you created a budget and deployed it. To learn more about Cost Management and Billing and Azure Resource Manager, continue on to the articles below.

- Read the [Cost Management and Billing](#) overview
- [Create budgets](#) in the Azure portal
- Learn more about [Azure Resource Manager](#)

# Use cost alerts to monitor usage and spending

Article • 04/06/2023

This article helps you understand and use Cost Management alerts to monitor your Azure usage and spending. Cost alerts are automatically generated based when Azure resources are consumed. Alerts show all active cost management and billing alerts together in one place. When your consumption reaches a given threshold, alerts are generated by Cost Management. There are three main types of cost alerts: budget alerts, credit alerts, and department spending quota alerts.

You can also [create a cost anomaly alert](#) to automatically get notified when an anomaly is detected.

## Required permissions for alerts

The following table shows how Cost Management alerts are used by each role. The behavior below is applicable to all Azure RBAC scopes.

Feature/Role	Owner	Contributor	Reader	Cost Management Reader	Cost Management Contributor
Alerts	Read, Update	Read, Update	Read only	Read only	Read, Update

## Budget alerts

Budget alerts notify you when spending, based on usage or cost, reaches or exceeds the amount defined in the [alert condition of the budget](#). Cost Management budgets are created using the Azure portal or the [Azure Consumption API](#).

In the Azure portal, budgets are defined by cost. Using the Azure Consumption API, budgets are defined by cost or by consumption usage. Budget alerts support both cost-based and usage-based budgets. Budget alerts are generated automatically whenever the budget alert conditions are met. You can view all cost alerts in the Azure portal. Whenever an alert is generated, it's shown in cost alerts. An alert email is also sent to the people in the alert recipients list of the budget.

If you have an Enterprise Agreement, you can [Create and edit budgets with PowerShell](#). Customers with a Microsoft Customer Agreement should use the [Budgets REST API](#) to

create budgets programmatically.

You can use the Budget API to send email alerts in a different language. For more information, see [Supported locales for budget alert emails](#).

## Credit alerts

Credit alerts notify you when your Azure Prepayment (previously called monetary commitment) is consumed. Azure Prepayment is for organizations with Enterprise Agreements. Credit alerts are generated automatically at 90% and at 100% of your Azure Prepayment credit balance. Whenever an alert is generated, it's reflected in cost alerts and in the email sent to the account owners.

## Department spending quota alerts

Department spending quota alerts notify you when department spending reaches a fixed threshold of the quota. Spending quotas are configured in the EA portal. Whenever a threshold is met it generates an email to department owners and is shown in cost alerts. For example, 50% or 75% of the quota.

## Supported alert features by offer categories

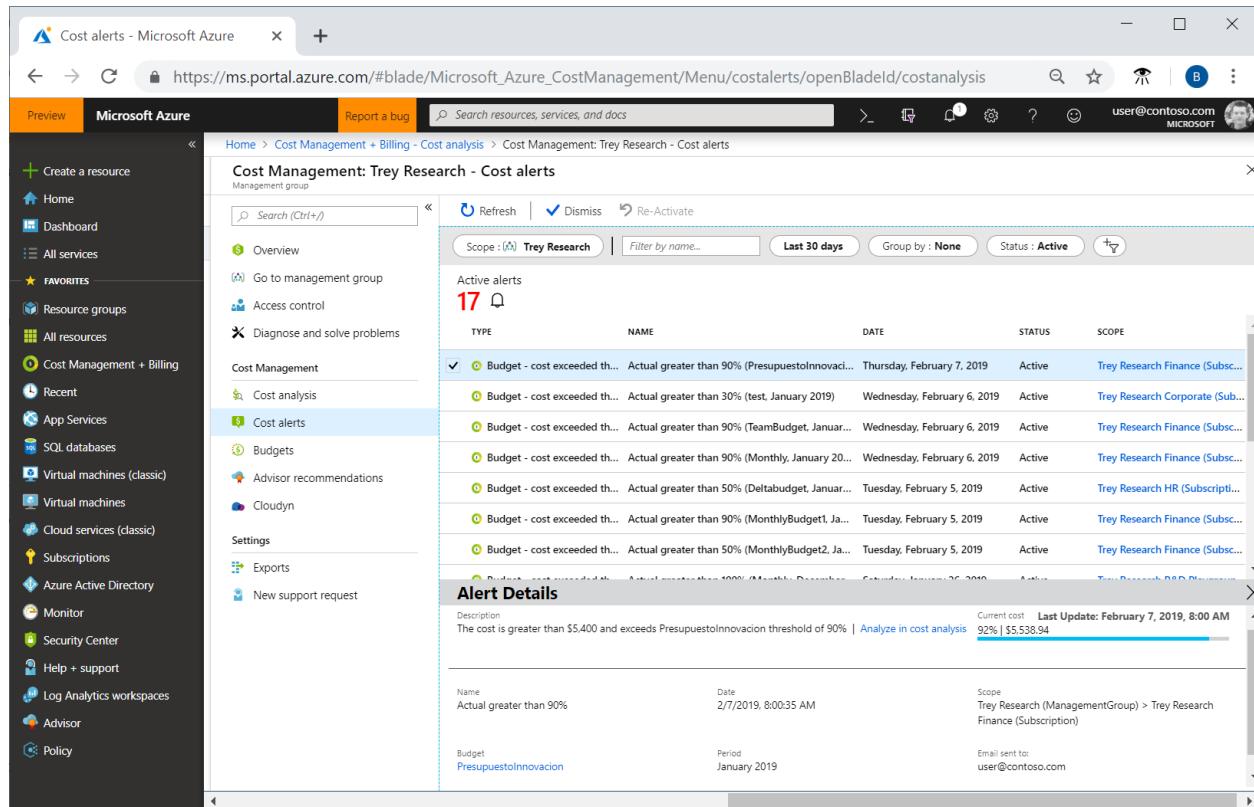
Support for alert types depends on the type of Azure account that you have (Microsoft offer). The following table shows the alert features that are supported by various Microsoft offers. You can view the full list of Microsoft offers at [Understand Cost Management data](#).

Alert type	Enterprise Agreement	Microsoft Customer Agreement	Web direct/Pay-As-You-Go
Budget	✓	✓	✓
Credit	✓	✗	✗
Department spending quota	✓	✗	✗

## View cost alerts

To view cost alerts, open the desired scope in the Azure portal and select **Budgets** in the menu. Use the **Scope** pill to switch to a different scope. Select **Cost alerts** in the menu.

For more information about scopes, see [Understand and work with scopes](#).



The screenshot shows the Microsoft Azure portal with the URL [https://ms.portal.azure.com/#blade/Microsoft\\_Azure\\_CostManagement/Menu/costalerts/openBladeId/costanalysis](https://ms.portal.azure.com/#blade/Microsoft_Azure_CostManagement/Menu/costalerts/openBladeId/costanalysis). The left sidebar is the Azure navigation menu. The main content area is titled "Cost Management: Trey Research - Cost alerts". It displays a list of 17 active alerts under the heading "Active alerts". Each alert row includes columns for TYPE, NAME, DATE, STATUS, and SCOPE. The alerts are all of type "Budget - cost exceeded threshold". The first alert's details are expanded, showing a description: "The cost is greater than \$5,400 and exceeds Presupuestolinovacion threshold of 90%". Below this, the "Alert Details" section provides specific information: Name (Actual greater than 90%), Date (2/7/2019, 8:00:35 AM), Scope (Trey Research (ManagementGroup) > Trey Research Finance (Subscription)), Budget (Presupuestolinovacion), Period (January 2019), and Email sent to (user@contoso.com). The status bar at the bottom indicates "Current cost: 92% | \$5,538.94" and "Last Update: February 7, 2019, 8:00 AM".

The total number of active and dismissed alerts appears on the cost alerts page.

All alerts show the alert type. A budget alert shows the reason why it was generated and the name of the budget it applies to. Each alert shows the date it was generated, its status, and the scope (subscription or management group) that the alert applies to.

Possible status includes **active** and **dismissed**. Active status indicates that the alert is still relevant. Dismissed status indicates that someone has marked the alert to set it as no longer relevant.

Select an alert from the list to view its details. Alert details show more information about the alert. Budget alerts include a link to the budget. If a recommendation is available for a budget alert, then a link to the recommendation is also shown. Budget, credit, and department spending quota alerts have a link to analyze in cost analysis where you can explore costs for the alert's scope. The following example shows spending for a department with alert details.

The screenshot shows the Azure portal's alert management interface. At the top, there are buttons for Refresh, Dismiss, and Re-Activate. The scope is set to 'Contoso (Demo) (8608480)'. Below this, a summary shows 3 Active alerts and 1 Dismissed alert. A table lists four alerts, with the last one being dismissed:

TYPE	NAME	DATE	STATUS	SCOPE
Department spending quota	Spend reached 100% (ACE)	Tuesday, January 15, 2019	Active	ACE (Department)
Department spending quota	Spend reached 100% (ACM)	Wednesday, December 12, 2018	Dismissed	ACM (Department)
Department spending quota	Spend reached 100% (DCX Program)	Wednesday, December 12, 2018	Active	DCX Program (Department)
Azure credit (system notification)	You have used over 100% of your azure credits	Wednesday, December 12, 2018	Active	Contoso (Demo) (8608480) (Billing account)

A modal window titled 'Alert Details' is open for the dismissed alert. It contains the following information:

- Description:** You have reached your department ACE spending quota 100% threshold. Spending quota: \$1,000 | [View in EA portal](#)
- Recommendation:** [Analyze in cost analysis](#)
- Current cost:** 2193% | \$21,931.59
- Last Update:** February 4, 2019, 5:00 PM
- Name:** Spend reached 100% (ACE)
- Date:** 1/15/2019, 5:02:51 PM
- Scope:** Contoso (Demo) (8608480) (BillingAccount) > ACE (Department)

When you view the details of a dismissed alert, you can reactivate it if manual action is needed. The following image shows an example.

The screenshot shows the Azure portal's alert management interface. At the top, there are buttons for Refresh, Dismiss (which has a red box around it), and Re-Activate. The scope is set to 'Trey Research Finance'. Below this, a summary shows 4 Active alerts and 1 Dismissed alert. A table lists five alerts, with the last one being dismissed:

TYPE	NAME
<input checked="" type="checkbox"/> Budget - cost exceeded threshold	Actual greater than 100% (PresupuestolInnovacion, January 2019)
<input type="checkbox"/> Budget - cost exceeded threshold	Actual greater than 50% (MonthlyBudget2, January 2019)
<input type="checkbox"/> Budget - cost exceeded threshold	Actual greater than 90% (MonthlyBudget1, January 2019)
<input type="checkbox"/> Budget - cost exceeded threshold	Actual greater than 90% (TeamBudget, January 2019)
<input checked="" type="checkbox"/> Budget - cost exceeded threshold	Actual greater than 90% (Monthly, January 2019)

## See also

- If you haven't already created a budget or set alert conditions for a budget, complete the [Create and manage budgets](#) tutorial.

# Manage costs with budgets

Article • 12/07/2022

Cost control is a critical component to maximizing the value of your investment in the cloud. There are several scenarios where cost visibility, reporting, and cost-based orchestration are critical to continued business operations. [Cost Management APIs](#) provide a set of APIs to support each of these scenarios. The APIs provide usage details, allowing you to view granular instance level costs.

Budgets are commonly used as part of cost control. Budgets can be scoped in Azure. For instance, you could narrow your budget view based on subscription, resource groups, or a collection of resources. In addition to using the budgets API to notify you via email when a budget threshold is reached, you can use [Azure Monitor action groups](#) to trigger an orchestrated set of actions resulting from a budget event.

A common budgets scenario for a customer running a non-critical workload could occur when they want to manage against a budget and also get to a predictable cost when looking at the monthly invoice. This scenario requires some cost-based orchestration of resources that are part of the Azure environment. In this scenario, a monthly budget of \$1000 for the subscription is set. Also, notification thresholds are set to trigger a few orchestrations. This scenario starts with an 80% cost threshold, which will stop all VMs in the resource group **Optional**. Then, at the 100% cost threshold, all VM instances will be stopped.

To configure this scenario, you'll complete the following actions by using the steps provided in each section of this tutorial.

These actions included in this tutorial allow you to:

- Create an Azure Automation Runbook to stop VMs by using webhooks.
- Create an Azure Logic App to be triggered based on the budget threshold value and call the runbook with the right parameters.
- Create an Azure Monitor Action Group that will be configured to trigger the Azure Logic App when the budget threshold is met.
- Create the budget with the wanted thresholds and wire it to the action group.

## Create an Azure Automation Runbook

[Azure Automation](#) is a service that enables you to script most of your resource management tasks and run those tasks as either scheduled or on-demand. As part of this scenario, you'll create an [Azure Automation runbook](#) that will be used to stop VMs.

You'll use the [Stop Azure V2 VMs](#) graphical runbook from the [gallery](#) to build this scenario. By importing this runbook into your Azure account and publishing it, you can stop VMs when a budget threshold is reached.

## Create an Azure Automation account

1. Sign in to the [Azure portal](#) with your Azure account credentials.
2. Select the **Create a resource** button found on the upper left corner of Azure.
3. Select **Management Tools > Automation**.

 **Note**

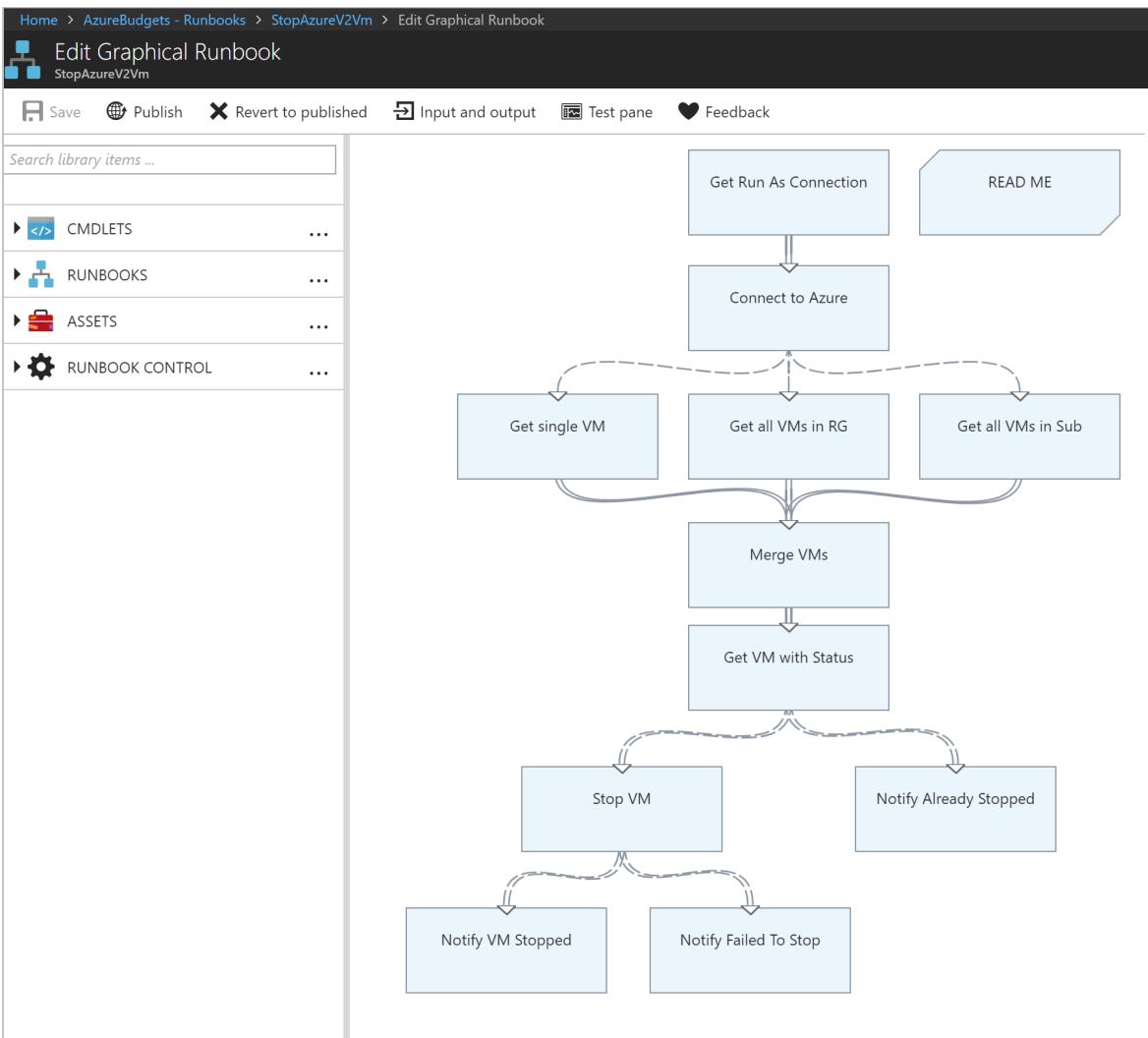
If you don't have an Azure account, you can create a [free account](#).

4. Enter your account information. For **Create Azure Run As account**, choose **Yes** to automatically enable the settings needed to simplify authentication to Azure.
5. When complete, select **Create**, to start the Automation account deployment.

## Import the Stop Azure V2 VMs runbook

Using an [Azure Automation runbook](#), import the [Stop Azure V2 VMs](#) graphical runbook from the gallery.

1. Sign in to the [Azure portal](#) with your Azure account credentials.
2. Open your Automation account by selecting **All services > Automation Accounts**. Then, select your Automation Account.
3. Select **Runbooks gallery** from the **Process Automation** section.
4. Set the **Gallery Source** to **Script Center** and select **OK**.
5. Locate and select the [Stop Azure V2 VMs](#) gallery item within the Azure portal.
6. Select **Import** to display the **Import** area and select **OK**. The runbook overview area will be displayed.
7. Once the runbook has completed the import process, select **Edit** to display the graphical runbook editor and publishing option.



8. Select **Publish** to publish the runbook and then select **Yes** when prompted. When you publish a runbook, you override any existing published version with the draft version. In this case, you've no published version because you've created the runbook. For more information about publishing a runbook, see [Create a graphical runbook](#).

## Create webhooks for the runbook

Using the [Stop Azure V2 VMs](#) graphical runbook, you create two Webhooks to start the runbook in Azure Automation through a single HTTP request. The first webhook invokes the runbook at an 80% budget threshold with the resource group name as a parameter, allowing the optional VMs to be stopped. Then, the second webhook invokes the runbook with no parameters (at 100%), which stops all remaining VM instances.

1. From the **Runbooks** page in the [Azure portal](#), select the **StopAzureV2Vm** runbook that displays the runbook's overview area.
2. Select **Webhook** at the top of the page to open the **Add Webhook** area.
3. Select **Create new webhook** to open the **Create a new webhook** area.

- Set the Name of the Webhook to **Optional**. The Enabled property must be **Yes**. You don't need to change the Expires value. For more information about Webhook properties, see [Webhook properties](#).
- Next to the URL value, select the copy icon to copy the URL of the webhook.

 **Important**

Save the URL of the webhook named **Optional** in a safe place. You'll use the URL later in this tutorial. For security reasons, once you create the webhook, you cannot view or retrieve the URL again.

- Select **OK** to create the new webhook.
- Select **Configure parameters and run settings** to view parameter values for the runbook.

 **Note**

If the runbook has mandatory parameters, then you are not able to create the webhook unless values are provided.

- Select **OK** to accept the webhook parameter values.
- Select **Create** to create the webhook.
- Next, follow the steps above to create a second webhook named **Complete**.

 **Important**

Be sure to save both webhook URLs to use later in this tutorial. For security reasons, once you create the webhook, you cannot view or retrieve the URL again.

You should now have two configured webhooks that are each available using the URLs that you saved.

Name	Expiration	Last Triggered	Status
Complete	7/18/2019, 4:06 PM		✓ Enabled
Optional	7/11/2019, 11:03 AM		✓ Enabled

You're now done with the Azure Automation setup. You can test the webhooks with a simple Postman test to validate that the webhook works. Next, you must create the Logic App for orchestration.

## Create an Azure Logic App for orchestration

Logic Apps helps you build, schedule, and automate processes as workflows so you can integrate apps, data, systems, and services across enterprises or organizations. In this scenario, the [Logic App](#) you create will do a little more than just call the automation webhook you created.

Budgets can be set up to trigger a notification when a specified threshold is met. You can provide multiple thresholds to be notified at and the Logic App will demonstrate the ability for you to perform different actions based on the threshold met. In this example, you'll set up a scenario where you get a couple of notifications, the first notification is for when 80% of the budget has been reached and the second notification is when 100% of the budget has been reached. The logic app will be used to shut down all VMs in the resource group. First, the **Optional** threshold will be reached at 80%, then the second threshold will be reached where all VMs in the subscription will be shut down.

Logic apps allow you to provide a sample schema for the HTTP trigger, but require you to set the **Content-Type** header. Because the action group doesn't have custom headers for the webhook, you must parse out the payload in a separate step. You'll use the **Parse** action and provide it with a sample payload.

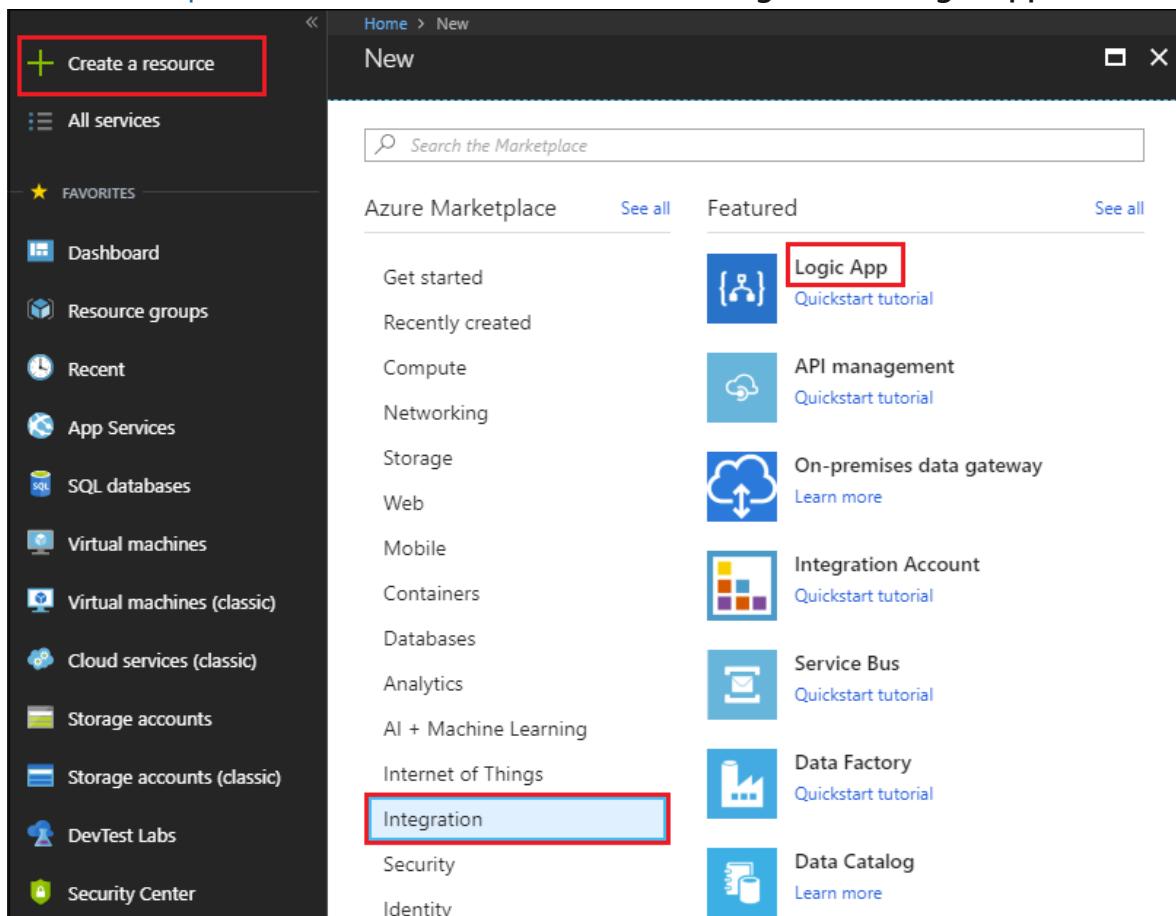
## Create the logic app

The logic app will perform several actions. The following list provides a high-level set of actions that the logic app will perform:

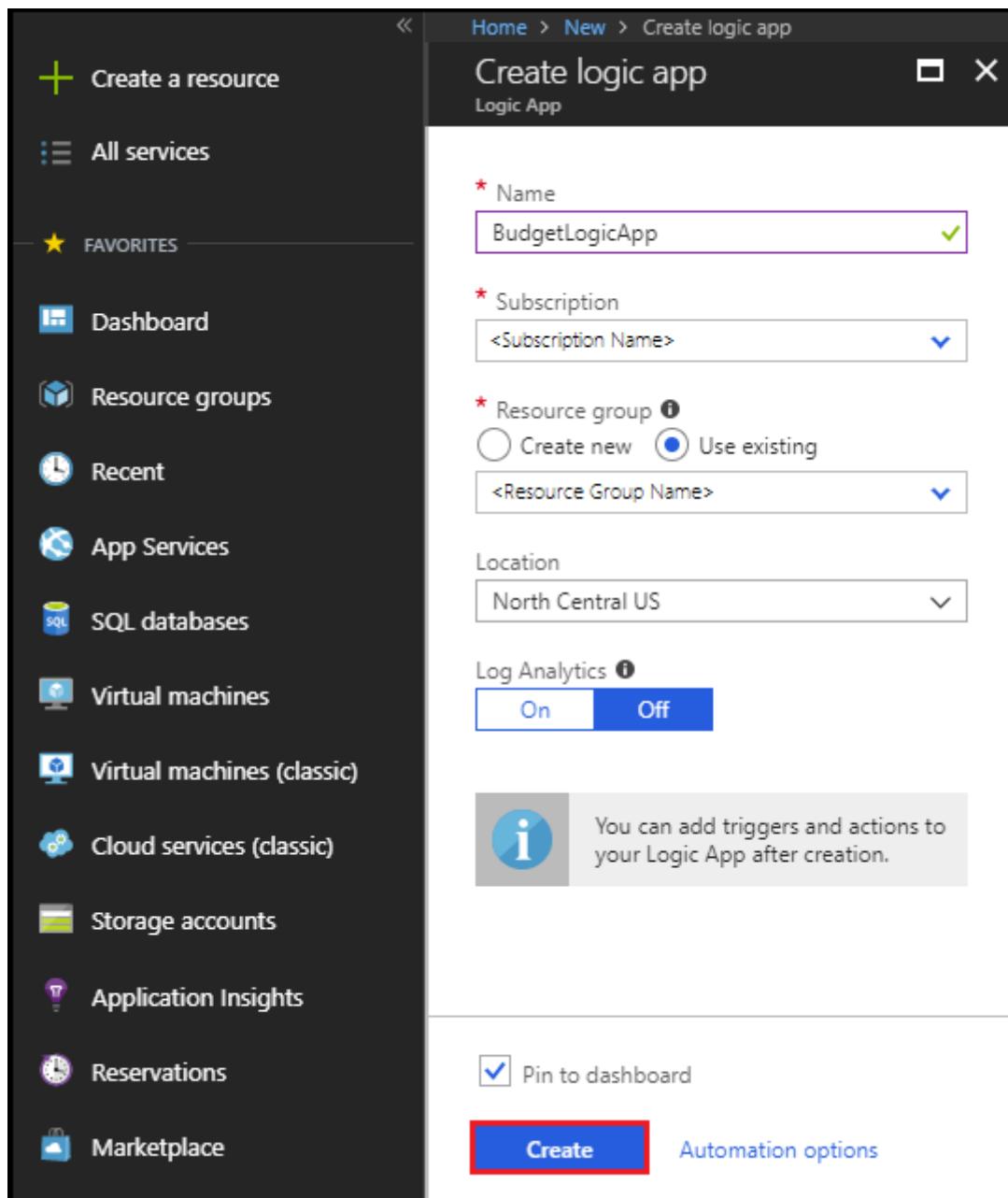
- Recognizes when an HTTP request is received
- Parse the passed in JSON data to determine the threshold value that has been reached
- Use a conditional statement to check whether the threshold amount has reached 80% or more of the budget range, but not greater than or equal to 100%.
  - If this threshold amount has been reached, send an HTTP POST using the webhook named **Optional**. This action will shut down the VMs in the "Optional" group.
- Use a conditional statement to check whether the threshold amount has reached or exceeded 100% of the budget value.
  - If the threshold amount has been reached, send an HTTP POST using the webhook named **Complete**. This action will shut down all remaining VMs.

The following steps are needed to create the logic app that will perform the above steps:

1. In the [Azure portal](#), select **Create a resource > Integration > Logic App**.



2. In the **Create logic app** area, provide the details need to create your logic app, select **Pin to dashboard**, and select **Create**.

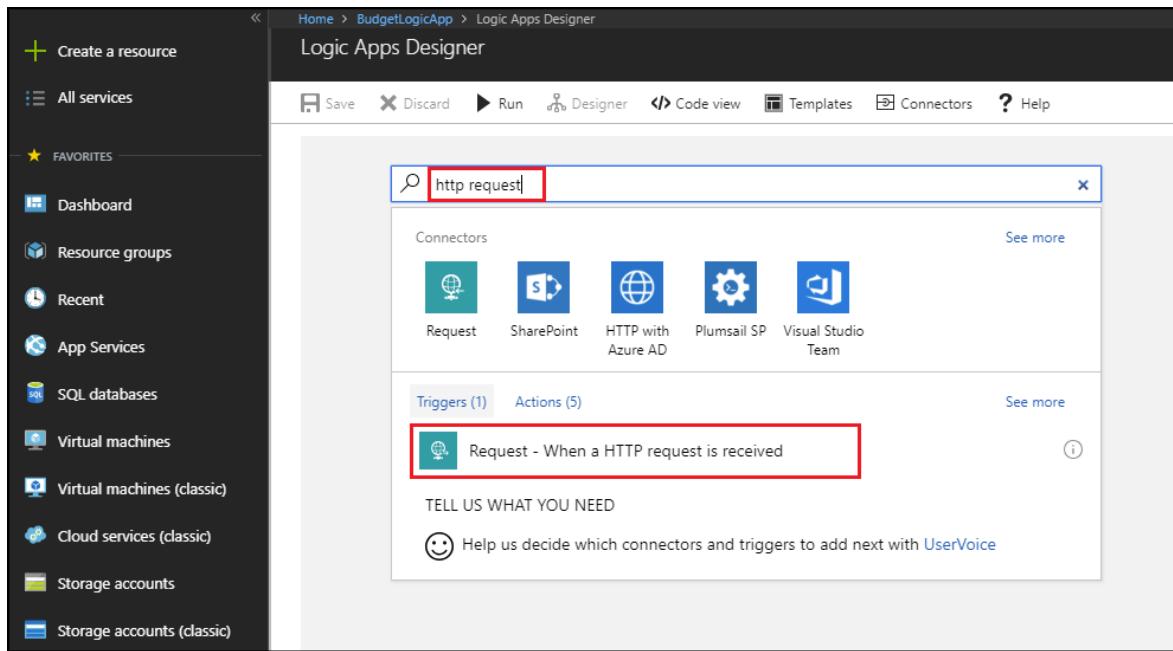


After Azure deploys your logic app, the **Logic Apps Designer** opens and shows an area with an introduction video and commonly used triggers.

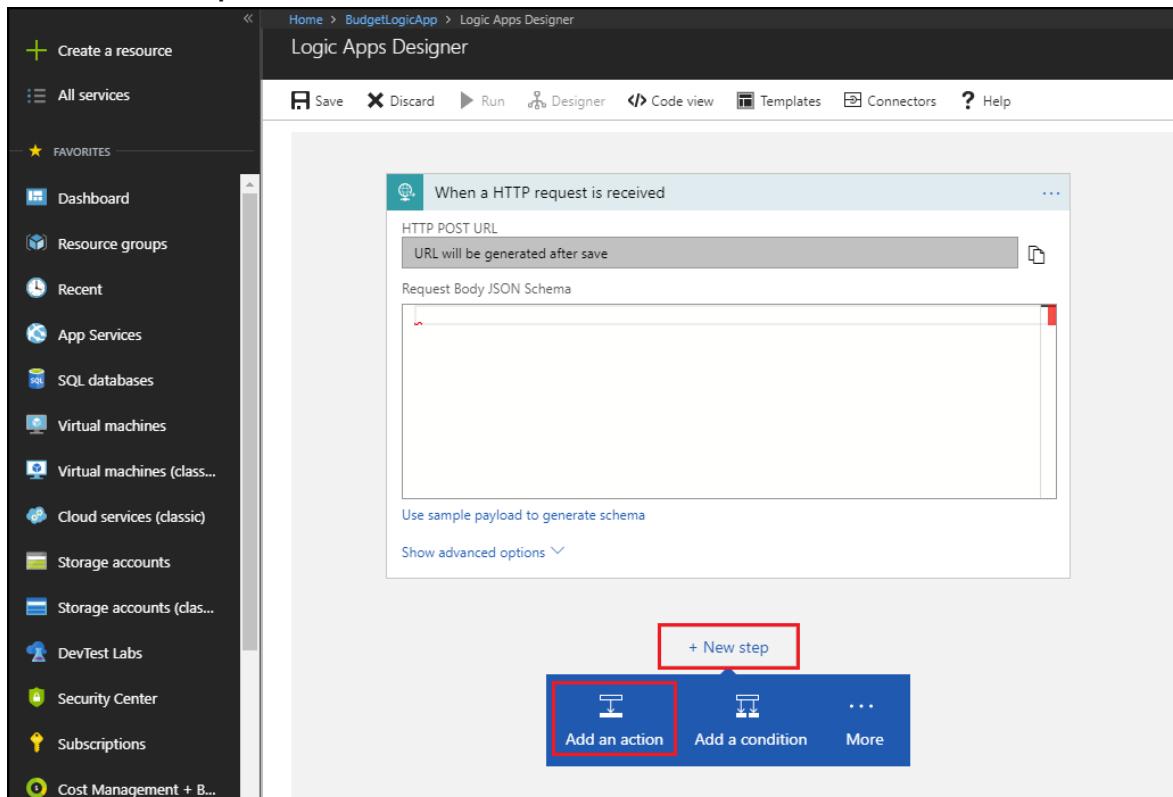
## Add a trigger

Every logic app must start with a trigger, which fires when a specific event happens or when a specific condition is met. Each time the trigger fires, the Logic Apps engine creates a logic app instance that starts and runs your workflow. Actions are all the steps that happen after the trigger.

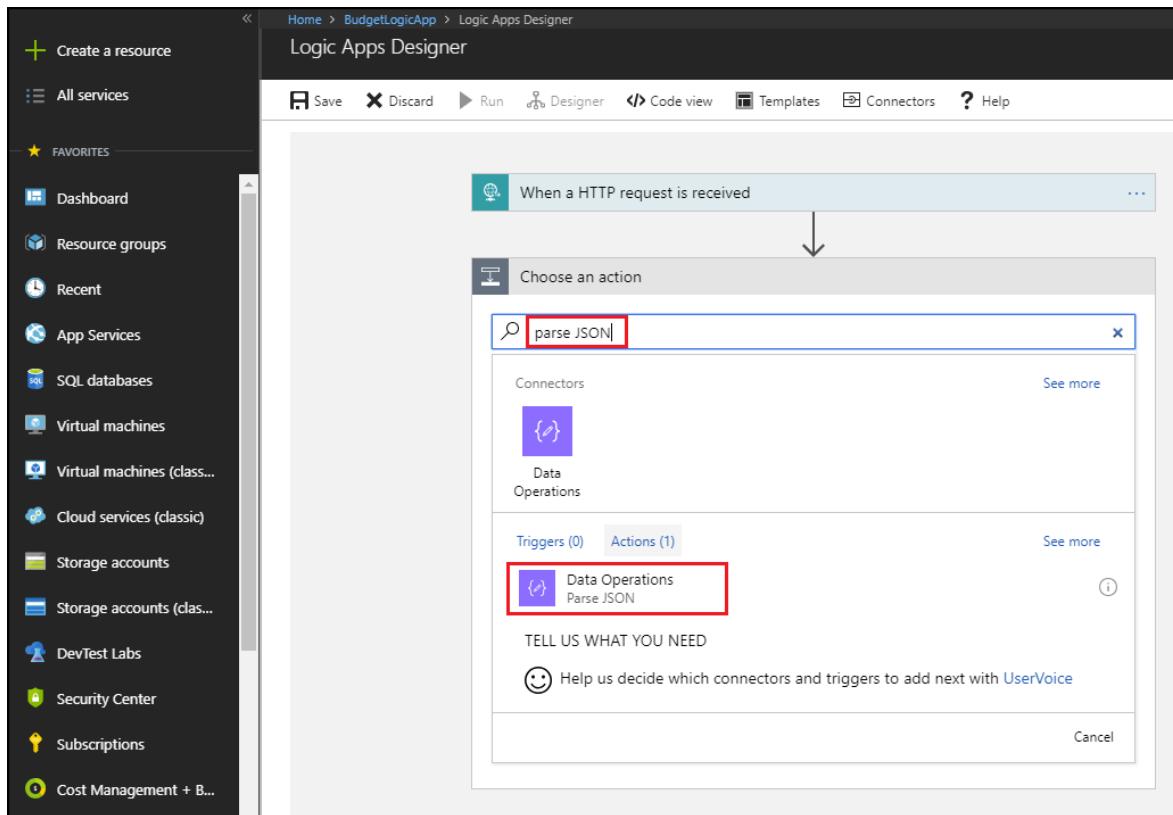
1. Under **Templates** of the **Logic Apps Designer** area, choose **Blank Logic App**.
2. Add a **trigger** by entering "http request" in the **Logic Apps Designer** search box to find and select the trigger named **Request – When an HTTP request is received**.



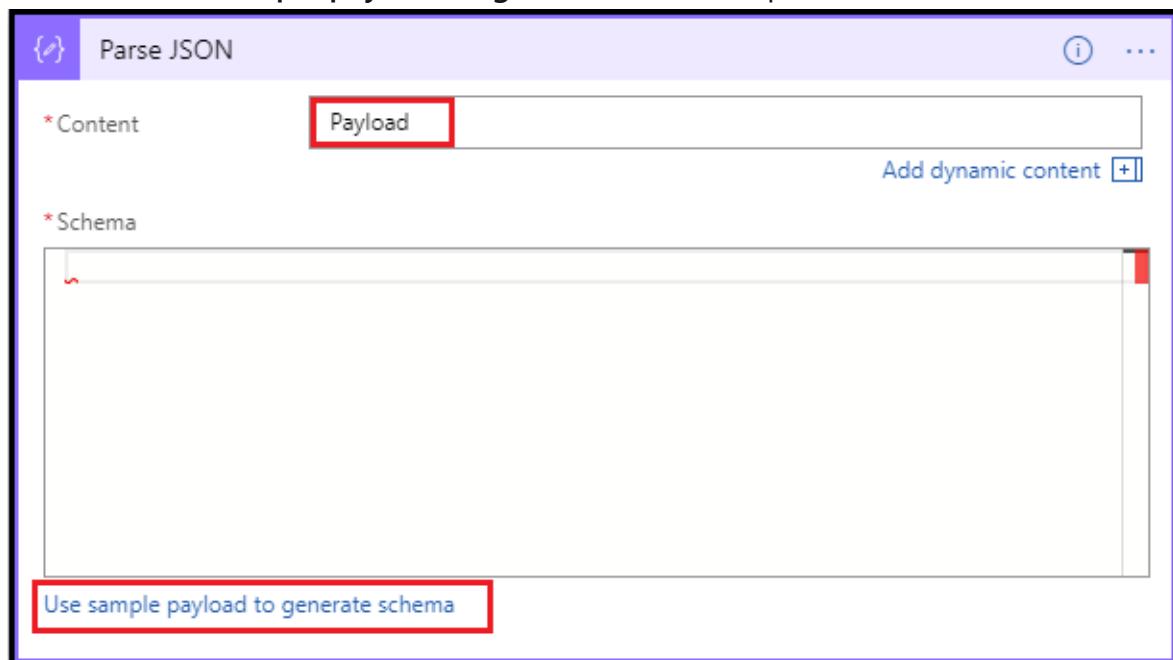
### 3. Select New step > Add an action.



### 4. Search for "parse JSON" in the Logic Apps Designer search box to find and select the Data Operations - Parse JSON action.



5. Enter "Payload" as the **Content** name for the Parse JSON payload or use the "Body" tag from dynamic content.
6. Select the **Use sample payload to generate schema** option in the Parse JSON box.



7. Paste the following JSON sample payload into the textbox:

```
{"schemaId": "AIP Budget Notification", "data": {"SubscriptionName": "CCM - Microsoft Azure Enterprise - 1", "SubscriptionId": "", "SpendingAmount": "100", "BudgetStartDate": "6/1/2018", "Budget": "50", "Unit": "USD", "BudgetCreator": "email@contoso.com", "BudgetName": "BudgetName", "BudgetType": "Cost", "ResourceGroup": "", "NotificationThresholdAmount": "0.8"}}
```

The text 'The' is also present at the end of the JSON block.

textbox will appear as:



8. Select Done.

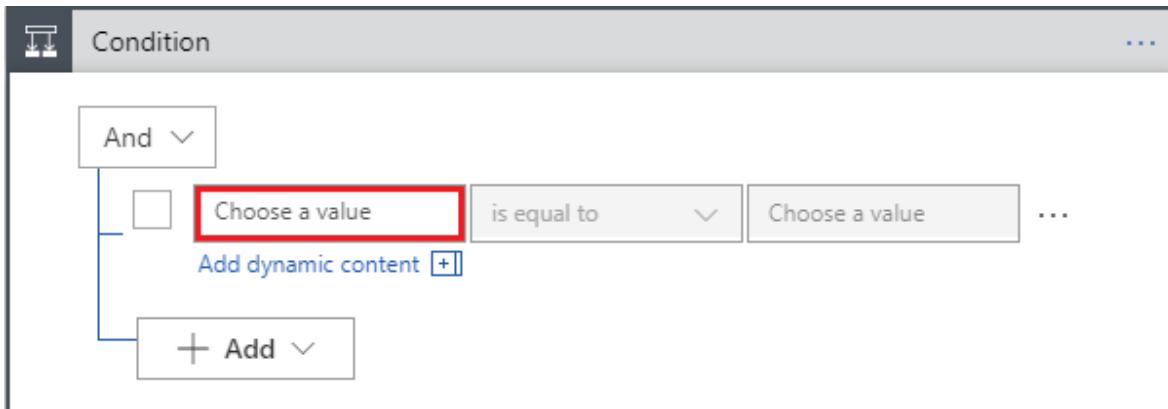
## Add the first conditional action

Use a conditional statement to check whether the threshold amount has reached 80% or more of the budget range, but not greater than or equal to 100%. If this threshold amount has been reached, send an HTTP POST using the webhook named **Optional**. This action will shut down the VMs in the **Optional** group.

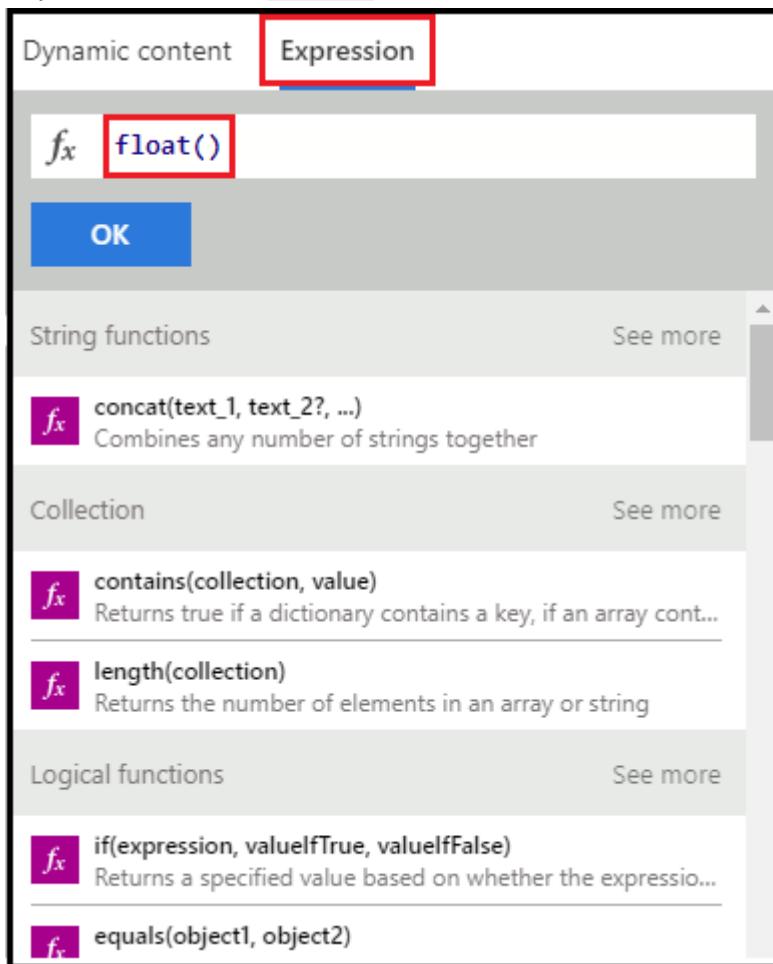
1. Select New step > Add a condition.



2. In the **Condition** box, select the textbox containing **Choose a value** to display a list of available values.



3. Select **Expression** at the top of the list and enter the following expression in the expression editor: `float()`



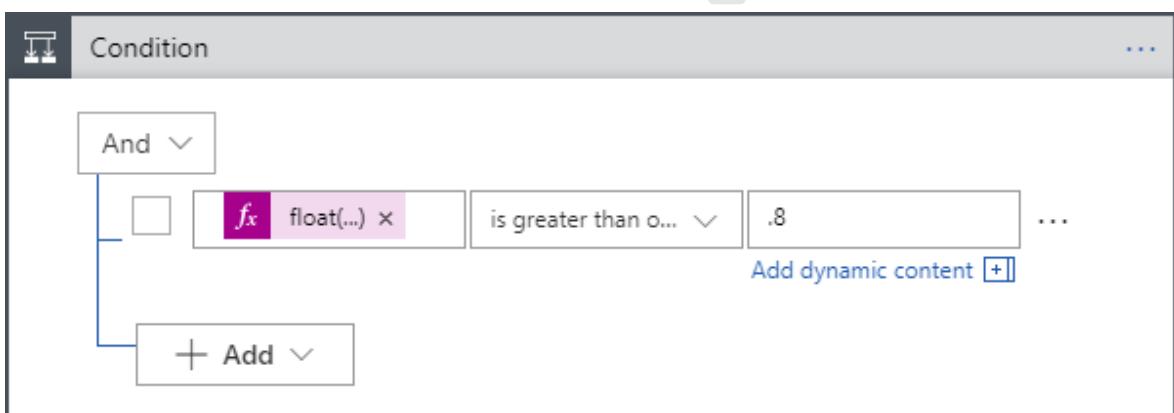
4. Select **Dynamic content**, place the cursor inside the parenthesis (), and select **NotificationThresholdAmount** from the list to populate the complete expression. The expression will be:

```
float(body('Parse_JSON')?['data']?['NotificationThresholdAmount'])
```

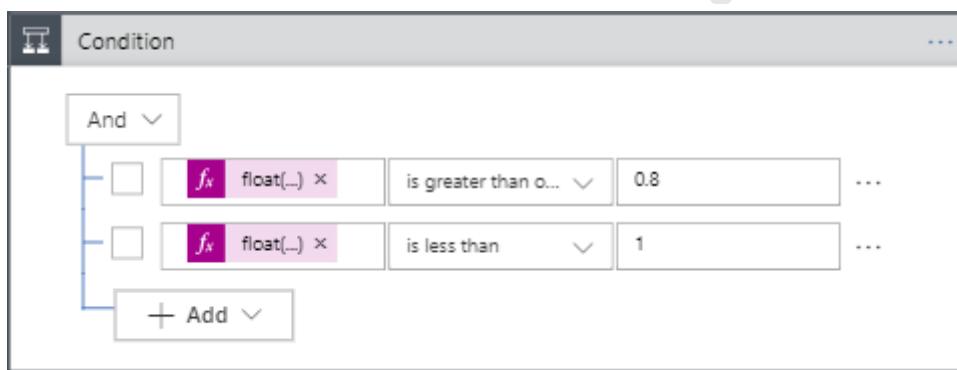
5. Select **OK** to set the expression.

6. Select **is greater than or equal to** in the dropdown box of the **Condition**.

7. In the **Choose a value** box of the condition, enter `.8`.



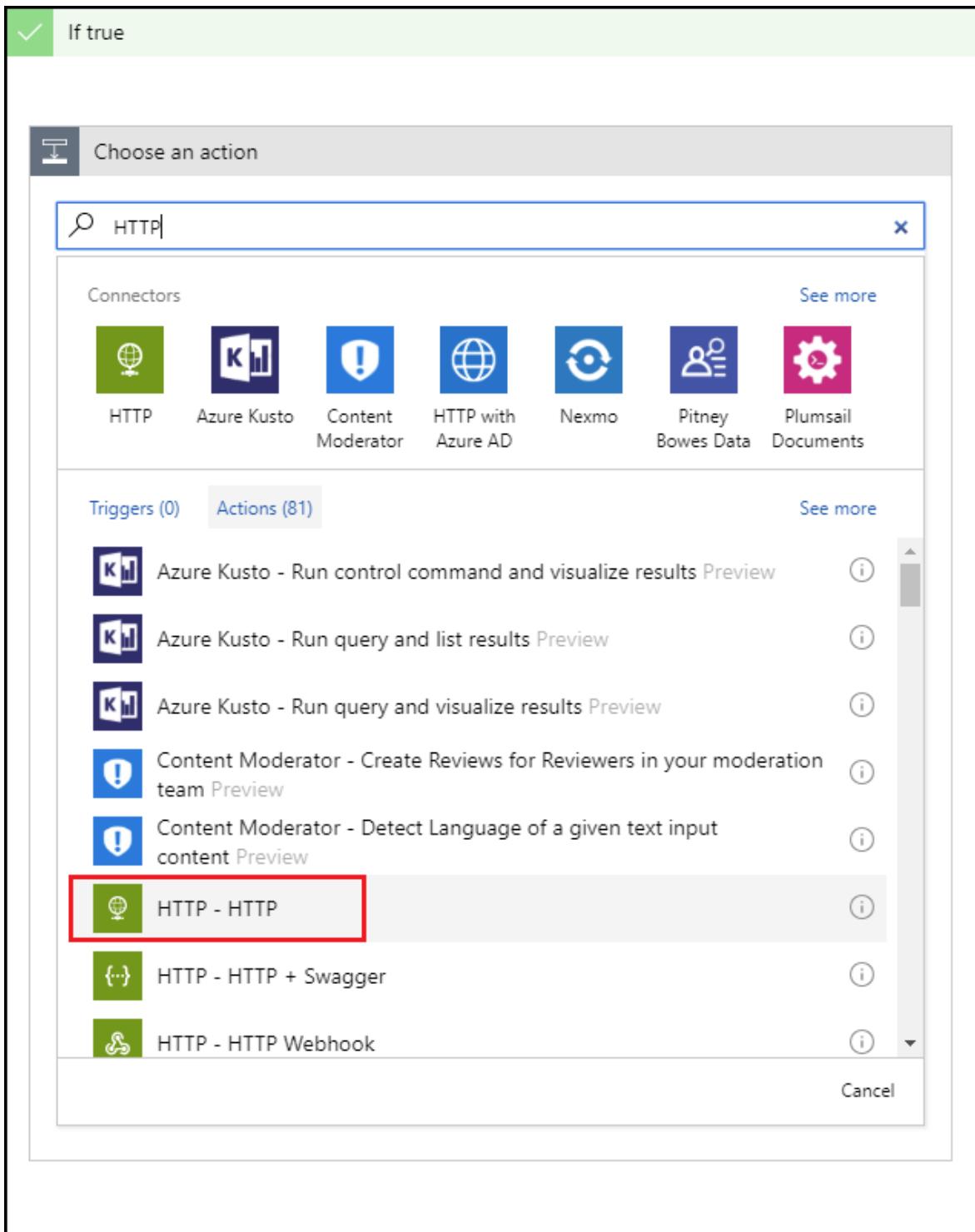
8. Select **Add > Add row** within the Condition box to add an additional part of the condition.
9. In the **Condition** box, select the textbox containing **Choose a value**.
10. Select **Expression** at the top of the list and enter the following expression in the expression editor: `float()`
11. Select **Dynamic content**, place the cursor inside the parenthesis (), and select **NotificationThresholdAmount** from the list to populate the complete expression.
12. Select **OK** to set the expression.
13. Select **is less than** in the dropdown box of the **Condition**.
14. In the **Choose a value** box of the condition, enter **1**.



15. In the **If true** box, select **Add an action**. You'll add an HTTP POST action that will shut down optional VMs.



16. Enter **HTTP** to search for the HTTP action and select the **HTTP – HTTP action**.



17. Select **Post** for the **Method** value.

18. Enter the URL for the webhook named **Optional** that you created earlier in this tutorial as the **Uri** value.

**HTTP**

* Method	post
* Uri	<webhook from Azure authomation>
Headers	Enter key   Enter value
Body	Enter request content

Show advanced options ▾

19. Select **Add an action** in the **If true** box. You'll add an email action that will send an email notifying the recipient that the optional VMs have been shut down.

20. Search for "send email" and select a *send email* action based on the email service you use.

Choose an action

x

Connectors

Office 365 Outlook	AWeber	Benchmark Email	FreshBooks	Gmail	MailChimp	Mandrill
--------------------	--------	-----------------	------------	-------	-----------	----------

Triggers (7) Actions (47) See more

- Office 365 Outlook - Send an email from a shared mailbox Preview
- Office 365 Outlook - Send email with options
- Outlook.com - Send an email
- Outlook.com - Send approval email
- Outlook.com - Delete email
- Outlook.com - Flag email Preview
- Outlook.com - Forward an email Preview
- Outlook.com - Get attachment

Cancel

For personal Microsoft accounts, select **Outlook.com**. For Azure work or school accounts, select **Office 365 Outlook**. If you don't already have a connection, you're asked to sign in to your email account. Logic Apps creates a connection to your email account. You'll need to allow the Logic App to access your email information.

 Microsoft

 Let this app access your info?

logic-apis-northcentralus.consent.azure-apim.net

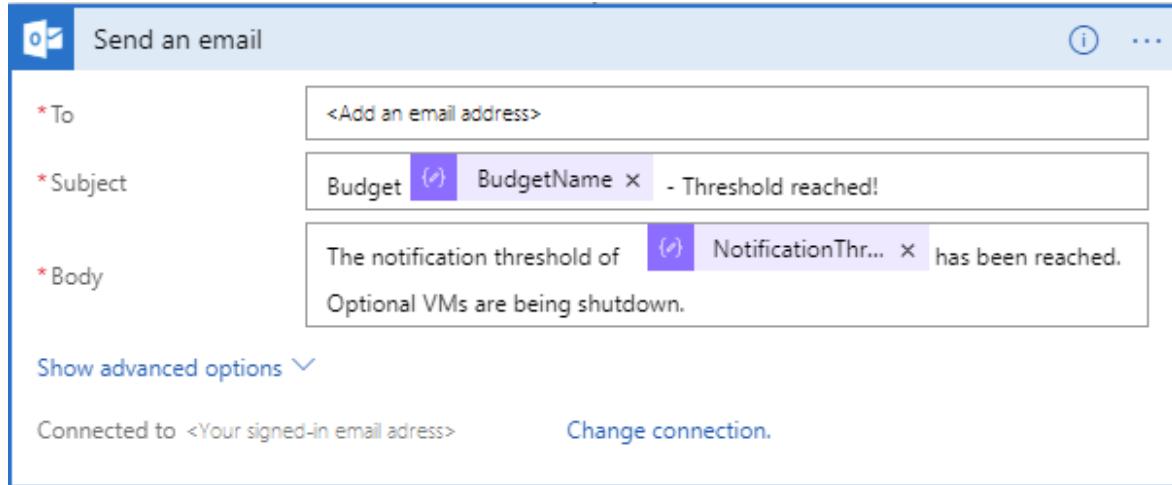
**Azure Logic Apps (North Central US) needs your permission to:**

-  **Read and write access to your mail**  
Azure Logic Apps (North Central US) will be able to read, update, create and delete email in your mailbox. Does not include permission to send mail.
-  **Send mail as you**  
Azure Logic Apps (North Central US) will be able to send mail as you.
-  **Have full access of your contacts**  
Azure Logic Apps (North Central US) will be able to read, update, create and delete contacts in your contact folders.
-  **Have full access to your calendars**  
Azure Logic Apps (North Central US) will be able to read, update, create and delete events in your calendars.
-  **Access your info anytime**  
Azure Logic Apps (North Central US) will be able to see and update your info, even when you're not using this app.

Accepting these permissions means that you allow this app to use your data as specified in their [terms of service](#) and [privacy statement](#). You can change these permissions at <https://microsoft.com/consent>. [Show details](#)

[Sign out](#)   [© 2018 Microsoft](#)   [Terms of Use](#)   [Privacy & Cookies](#)

21. Add the **To**, **Subject**, and **Body** text for the email that notifies the recipient that the optional VMs have been shut down. Use the **BudgetName** and the **NotificationThresholdAmount** dynamic content to populate the subject and body fields.



## Add the second conditional action

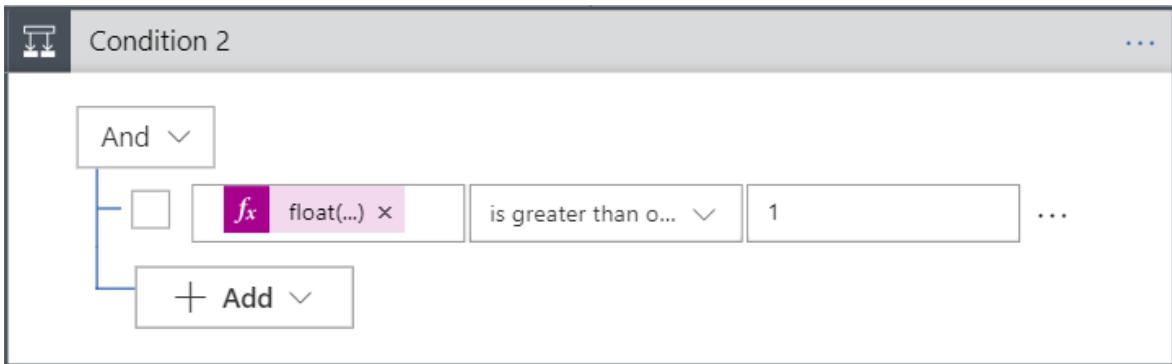
Use a conditional statement to check whether the threshold amount has reached or exceeded 100% of the budget value. If the threshold amount has been reached, send an HTTP POST using the webhook named **Complete**. This action will shut down all remaining VMs.

1. Select **New step > Add a Condition**.



2. In the **Condition** box, select the textbox containing `Choose a value` to display a list of available values.
3. Select **Expression** at the top of the list and enter the following expression in the expression editor: `float()`
4. Select **Dynamic content**, place the cursor inside the parenthesis (), and select **NotificationThresholdAmount** from the list to populate the complete expression. The expression will resemble:  
`float(body('Parse_JSON')?['data']?['NotificationThresholdAmount'])`
5. Select **OK** to set the expression.
6. Select **is greater than or equal to** in the dropdown box of the **Condition**.

7. In the Choose a value box for the condition, enter 1.



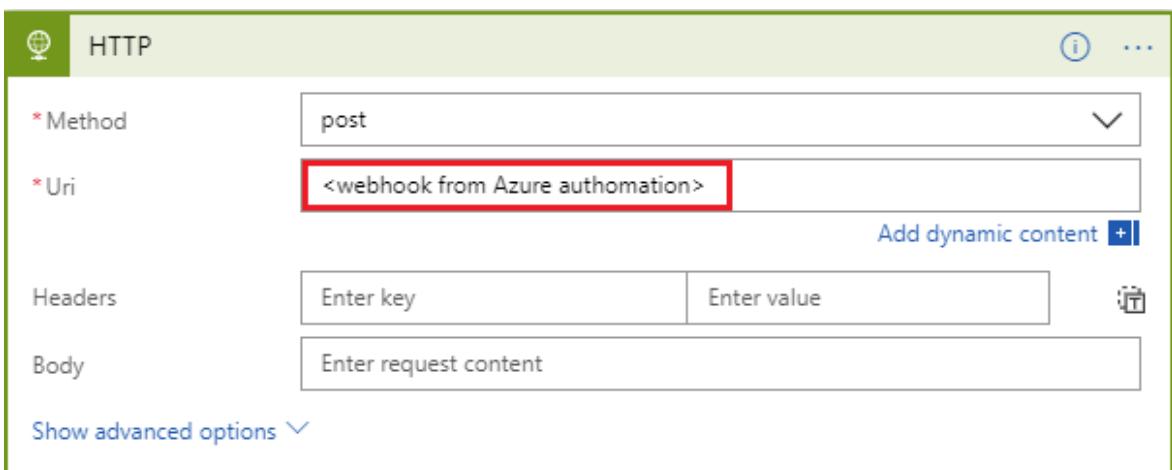
8. In the If true box, select Add an action. You'll add an HTTP POST action that will shut down all the remaining VMs.



9. Enter HTTP to search for the HTTP action and select the HTTP – HTTP action.

10. Select Post as the Method value.

11. Enter the URL for the webhook named Complete that you created earlier in this tutorial as the Uri value.

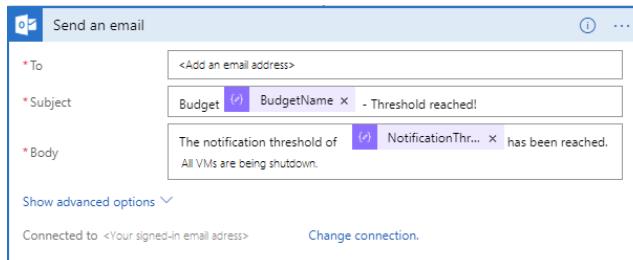


12. Select Add an action in the If true box. You'll add an email action that will send an email notifying the recipient that the remaining VMs have been shut down.

13. Search for "send email" and select a *send email* action based on the email service you use.

14. Add the To, Subject, and Body text for the email that notifies the recipient that the optional VMs have been shut down. Use the **BudgetName** and the **NotificationThresholdAmount** dynamic content to populate the subject and body

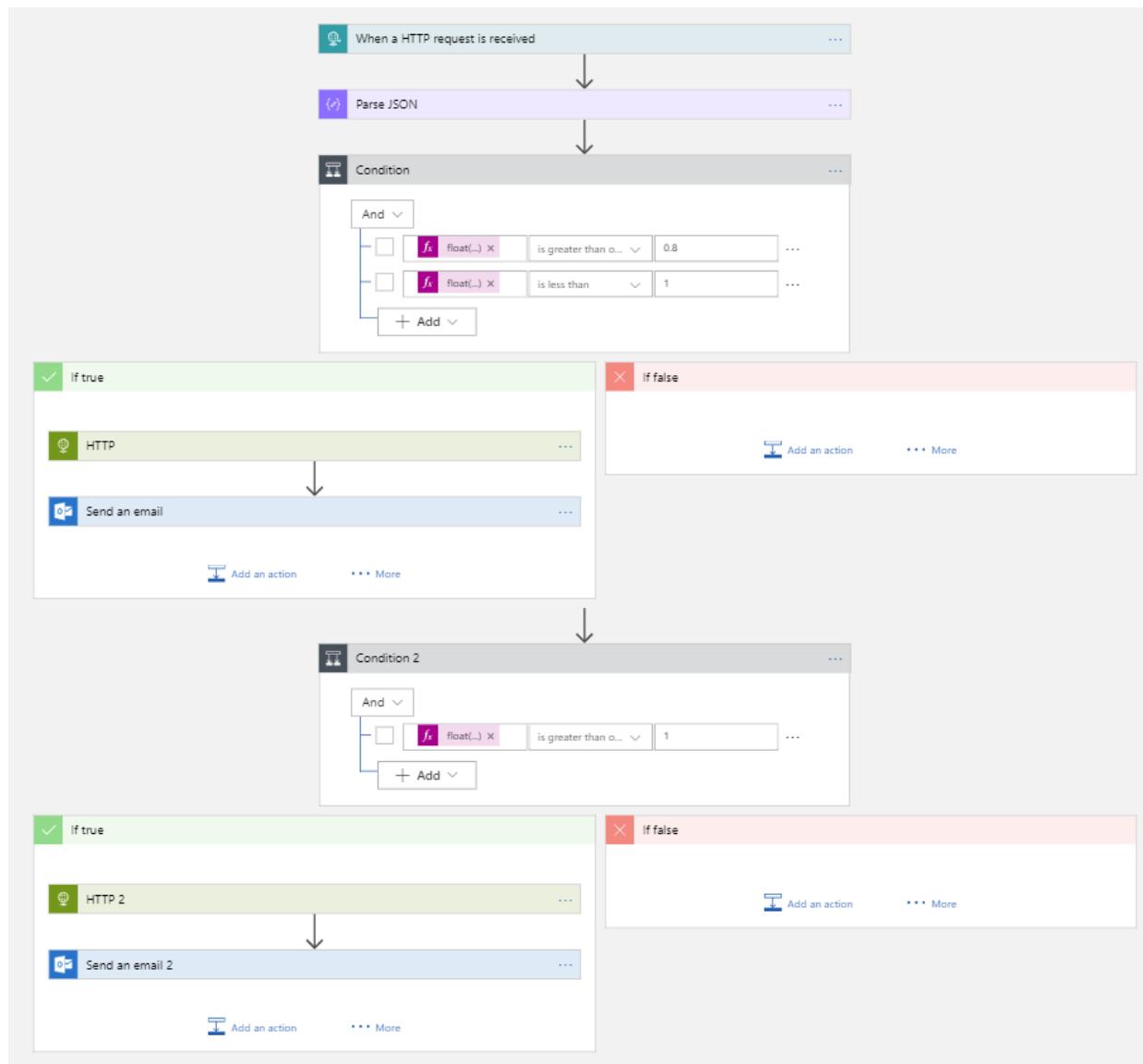
fields.



15. Select **Save** at the top of the Logic App Designer area.

## Logic App summary

Here's what your Logic App looks like once you're done. In the most basic of scenarios where you don't need any threshold-based orchestration, you could directly call the automation script from **Monitor** and skip the **Logic App** step.



When you saved your logic app, a URL was generated that you'll be able to call. You'll use this URL in the next section of this tutorial.

# Create an Azure Monitor Action Group

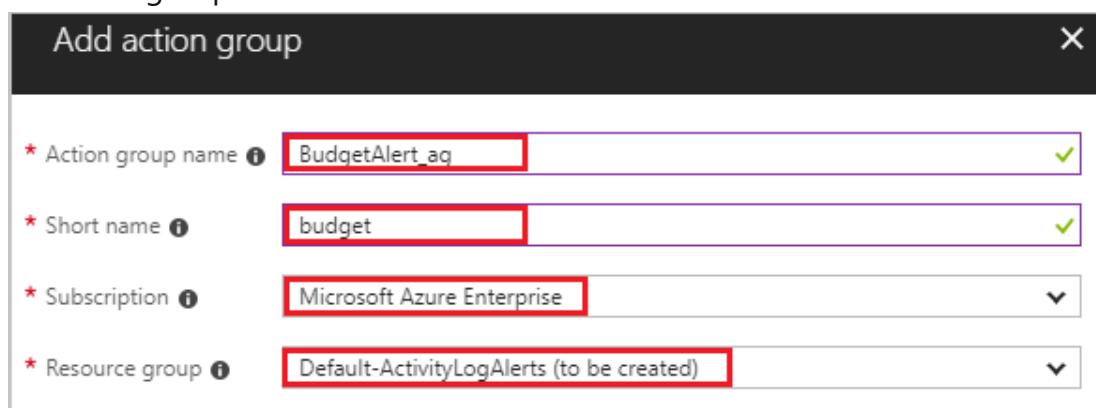
An action group is a collection of notification preferences that you define. When an alert is triggered, a specific action group can receive the alert by being notified. An Azure alert proactively raises a notification based on specific conditions and provides the opportunity to take action. An alert can use data from multiple sources, including metrics and logs.

Action groups are the only endpoint that you'll integrate with your budget. You can set up notifications in a number of channels, but for this scenario you'll focus on the Logic App you created earlier in this tutorial.

## Create an action group in Azure Monitor

When you create the action group, you'll point to the Logic App that you created earlier in this tutorial.

1. If you are not already signed-in to the [Azure portal](#), sign in and select **All services > Monitor**.
2. Select **Alerts** then select **Manage actions**.
3. Select **Add an action group** from the **Action groups** area.
4. Add and verify the following items:
  - Action group name
  - Short name
  - Subscription
  - Resource group



5. Within the **Add action group** pane, add a **LogicApp** action. Name the action **Budget-BudgetLA**. In the **Logic App** pane, select the **Subscription** and the **Resource group**. Then, select the **Logic app** that you created earlier in this tutorial.
6. Select **OK** to set the Logic App. Then, select **OK** in the **Add action group** pane to create the action group.

You're done with all the supporting components needed to effectively orchestrate your budget. Now all you need to do is create the budget and configure it to use the action group you created.

## Create the budget

You can create a budget in the Azure portal using the [Budget feature](#) in Cost Management. Or, you can create a budget using REST APIs, PowerShell cmdlets, or use the CLI. The following procedure uses the REST API. Before calling the REST API, you'll need an authorization token. To create an authorization token, you can use the [ARMClient](#) project. The **ARMClient** allows you to authenticate yourself to the Azure Resource Manager and get a token to call the APIs.

## Create an authentication token

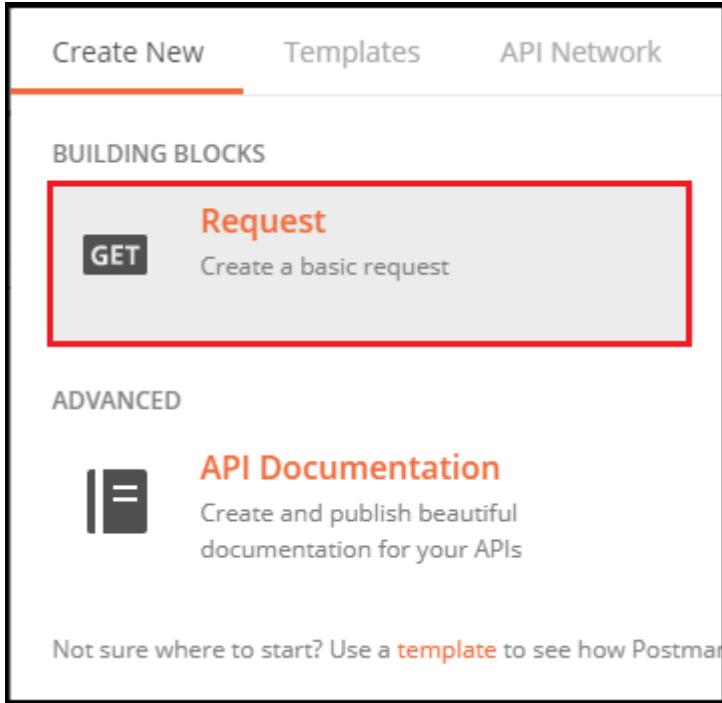
1. Navigate to the [ARMClient](#) project on GitHub.
2. Clone the repo to get a local copy.
3. Open the project in Visual Studio and build it.
4. Once the build is successful, the executable should be in the `\bin\debug` folder.
5. Run the ARMClient. Open a command prompt and navigate to the `\bin\debug` folder from the project root.
6. To sign in and authenticate, enter the following command at the command prompt:  
`ARMClient login prod`
7. Copy the **subscription guid** from the output.
8. To copy an authorization token to your clipboard, enter the following command at the command prompt, but sure to use the copied subscription ID from the step above:  
`ARMClient token <subscription GUID from previous step>`

Once you have completed the step above, you'll see:  
**Token copied to clipboard successfully.**
9. Save the token to be used for steps in the next section of this tutorial.

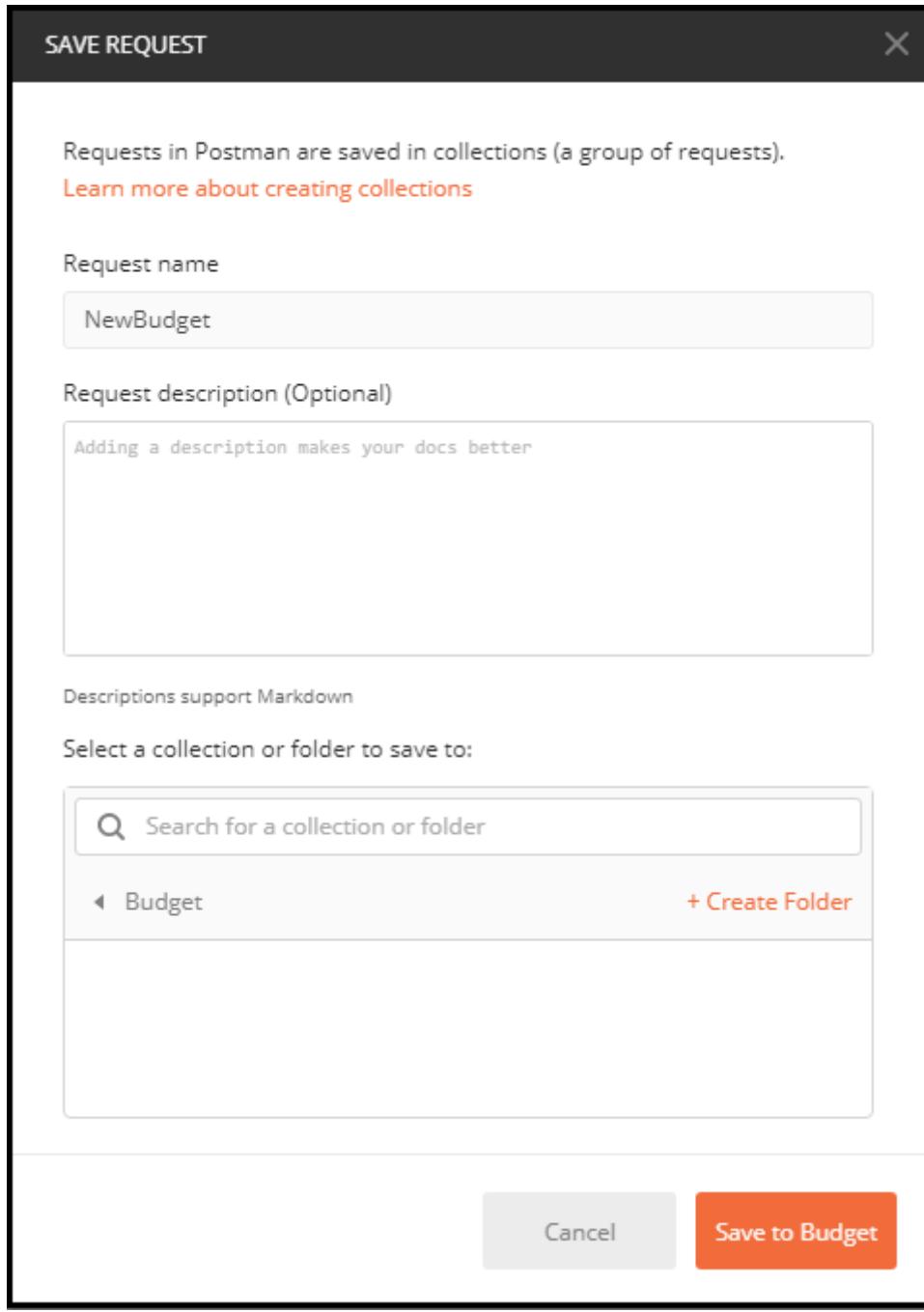
## Create the Budget

Next, you'll configure **Postman** to create a budget by calling the Azure Consumption REST APIs. Postman is an API Development environment. You'll import environment and collection files into Postman. The collection contains grouped definitions of HTTP requests that call Azure Consumption REST APIs. The environment file contains variables that are used by the collection.

1. Download and open the [Postman REST client](#) to execute the REST APIs.
2. In Postman, create a new request.



3. Save the new request as a collection, so that the new request has nothing on it.



4. Change the request from a `Get` to a `Put` action.

5. Modify the following URL by replacing `{subscriptionId}` with the **Subscription ID** that you used in the previous section of this tutorial. Also, modify the URL to include "SampleBudget" as the value for `{budgetName}`:

```
https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.Consumption/budgets/{budgetName}?api-version=2018-03-31
```

6. Select the **Headers** tab within Postman.

7. Add a new **Key** named "Authorization".

8. Set the **Value** to the token that was created using the ArmClient at the end of the last section.

9. Select **Body** tab within Postman.

10. Select the **raw** button option.

11. In the textbox, paste in the below sample budget definition, however you must replace the `subscriptionID`, `resourcegroupname`, and `actiongroupname` parameters with your subscription ID, a unique name for your resource group, and the action group name you created in both the URL and the request body:

```
{
 "properties": {
 "category": "Cost",
 "amount": 100.00,
 "timeGrain": "Monthly",
 "timePeriod": {
 "startDate": "2018-06-01T00:00:00Z",
 "endDate": "2018-10-31T00:00:00Z"
 },
 "filters": {},
 "notifications": {
 "Actual_GreaterThan_80_Percent": {
 "enabled": true,
 "operator": "GreaterThan",
 "threshold": 80,
 "contactEmails": [],
 "contactRoles": [],
 "contactGroups": [
 "/subscriptions/{subscriptionid}/resourceGroups/{resourcegroupname}/providers/microsoft.insights/actionGroups/{actiongroupname}"
]
 },
 "Actual_EqualTo_100_Percent": {
 "operator": "EqualTo",
 "threshold": 100,
 "contactGroups": [
 "/subscriptions/{subscriptionid}/resourceGroups/{resourcegroupname}/providers/microsoft.insights/actionGroups/{actiongroupname}"
]
 }
 }
}
```

12. Press **Send** to send the request.

You now have all the pieces you need to call the [budgets API](#). The budgets API reference has additional details on the specific requests, including:

- **budgetName** - Multiple budgets are supported. Budget names must be unique.
- **category** - Must be either **Cost** or **Usage**. The API supports both cost and usage budgets.
- **timeGrain** - A monthly, quarterly, or yearly budget. The amount resets at the end of the period.
- **filters** - Filters allow you to narrow the budget to a specific set of resources within the selected scope. For example, a filter could be a collection of resource groups for a subscription level budget.
- **notifications** – Determines the notification details and thresholds. You can set up multiple thresholds and provide an email address or an action group to receive a notification.

## Summary

By using this tutorial, you learned:

- How to create an Azure Automation Runbook to stop VMs.
- How to create an Azure Logic App that is triggered based on the budget threshold values and call the related runbook with the right parameters.
- How to create an Azure Monitor Action Group that was configured to trigger the Azure Logic App when the budget threshold is met.
- How to create the budget with the desired thresholds and wire it to the action group.

You now have a fully functional budget for your subscription that will shut down your VMs when you reach your configured budget thresholds.

## Next steps

- For more information about Azure billing scenarios, see [Billing and cost management automation scenarios](#).

# Alerts

Reference

Service: Cost Management

API Version: 2023-03-01

## Operations

<a href="#">Dismiss</a>	Dismisses the specified alert
<a href="#">Get</a>	Gets the alert for the scope by alert ID.
<a href="#">List</a>	Lists the alerts for scope defined.
<a href="#">List External</a>	Lists the Alerts for external cloud provider type defined.

# Scheduled Actions

Reference

Service: Cost Management

API Version: 2023-03-01

## Operations

<a href="#">Check Name Availability</a>	Checks availability and correctness of the name for a scheduled action.
<a href="#">Check Name Availability By Scope</a>	Checks availability and correctness of the name for a scheduled action within the given scope.
<a href="#">Create Or Update</a>	Create or update a private scheduled action.
<a href="#">Create Or Update By Scope</a>	Create or update a shared scheduled action within the given scope.
<a href="#">Delete</a>	Delete a private scheduled action.
<a href="#">Delete By Scope</a>	Delete a scheduled action within the given scope.
<a href="#">Get</a>	Get the private scheduled action by name.
<a href="#">Get By Scope</a>	Get the shared scheduled action from the given scope by name.
<a href="#">List</a>	List all private scheduled actions.
<a href="#">List By Scope</a>	List all shared scheduled actions within the given scope.
<a href="#">Run</a>	Processes a private scheduled action.
<a href="#">Run By Scope</a>	Runs a shared scheduled action within the given scope.

# Microsoft.Consumption budgets

Article • 12/28/2022

## Bicep resource definition

The budgets resource type is an [extension resource](#), which means you can apply it to another resource.

Use the `scope` property on this resource to set the scope for this resource. See [Set scope on extension resources in Bicep](#).

Valid deployment scopes for the budgets resource are:

- **Resource groups** - See [resource group deployment commands](#)
- **Subscriptions** - See [subscription deployment commands](#)

For a list of changed properties in each API version, see [change log](#).

## Resource format

To create a Microsoft.Consumption/budgets resource, add the following Bicep to your template.

Bicep

```
resource symbolicname 'Microsoft.Consumption/budgets@2021-10-01' = {
 name: 'string'
 scope: resourceSymbolicName
 eTag: 'string'
 properties: {
 amount: int
 category: 'Cost'
 filter: {
 and: [
 {
 dimensions: {
 name: 'string'
 operator: 'In'
 values: [
 'string'
]
 }
 tags: {
 name: 'string'
 operator: 'In'
 values: [

```

```

 'string'
]
}
]
dimensions: {
 name: 'string'
 operator: 'In'
 values: [
 'string'
]
}
tags: {
 name: 'string'
 operator: 'In'
 values: [
 'string'
]
}
}
notifications: {}
timeGrain: 'string'
timePeriod: {
 endDate: 'string'
 startDate: 'string'
}
}
}
}

```

## Property values

### budgets

Name	Description	Value
name	The resource name	<p>string (required)</p> <p>Character limit: 1-63</p> <p>Valid characters: Alphanumerics, hyphens, and underscores.</p>
scope	Use when creating an extension resource at a scope that is different than the deployment scope.	<p>Target resource</p> <p>For Bicep, set this property to the symbolic name of the resource to apply the <a href="#">extension resource</a>.</p>

Name	Description	Value
eTag	eTag of the resource. To handle concurrent update scenario, this field will be used to determine whether the user is updating the latest version or not.	string
properties	The properties of the budget.	<a href="#">BudgetProperties</a>

## BudgetProperties

Name	Description	Value
amount	The total amount of cost to track with the budget	int (required)
category	The category of the budget, whether the budget tracks cost or usage.	'Cost' (required)
filter	May be used to filter budgets by user-specified dimensions and/or tags.	<a href="#">BudgetFilter</a>
notifications	Dictionary of notifications associated with the budget. Budget can have up to five notifications.	object
timeGrain	The time covered by a budget. Tracking of the amount will be reset based on the time grain. BillingMonth, BillingQuarter, and BillingAnnual are only supported by WD customers	'Annually' 'BillingAnnual' 'BillingMonth' 'BillingQuarter' 'Monthly' 'Quarterly' (required)
timePeriod	Has start and end date of the budget. The start date must be first of the month and should be less than the end date.  Budget start date must be on or after June 1, 2017. Future start date should not be more than twelve months. Past start date should be selected within the timegrain period. There are no restrictions on the end date.	<a href="#">BudgetTimePeriod</a> (required)

## BudgetFilter

Name	Description	Value
and	The logical "AND" expression. Must have at least 2 items.	<a href="#">BudgetFilterProperties[]</a>
dimensions	Has comparison expression for a dimension	<a href="#">BudgetComparisonExpression</a>

Name	Description	Value
tags	Has comparison expression for a tag	object

## BudgetFilterProperties

Name	Description	Value
dimensions	Has comparison expression for a dimension	<a href="#">BudgetComparisonExpression</a>
tags	Has comparison expression for a tag	object

## BudgetComparisonExpression

Name	Description	Value
name	The name of the column to use in comparison.	string (required)
operator	The operator to use for comparison.	'In' (required)
values	Array of values to use for comparison	string[] (required)

## BudgetTimePeriod

Name	Description	Value
endDate	The end date for the budget. If not provided, we default this to 10 years from the start date.	string
startDate	The start date for the budget.	string (required)

## Quickstart templates

The following quickstart templates deploy this resource type.

Template	Description
<a href="#">Create a Budget</a>	This template shows how to create a budget under a subscription.



Deploy to Azure

Template	Description
<a href="#">Create a Budget with Filter</a>	This template shows how to create a budget under a subscription.  <a href="#"> Deploy to Azure</a>
<a href="#">Create a Simple Budget</a>	This template shows how to create a budget under a subscription.  <a href="#"> Deploy to Azure</a>

# Tutorial: Optimize costs from recommendations

Article • 12/08/2022

Cost Management works with Azure Advisor to provide cost optimization recommendations. Azure Advisor helps you optimize and improve efficiency by identifying idle and underutilized resources. This tutorial walks you through an example where you identify underutilized Azure resources and then you take action to reduce costs.

Watch the video [Optimizing cloud investments in Cost Management](#) to learn more about using Advisor to optimize your costs. To watch other videos, visit the [Cost Management YouTube channel](#).

<https://www.youtube-nocookie.com/embed/cSNPoAb-TNc>

In this tutorial, you learn how to:

- ✓ View cost optimization recommendations to view potential usage inefficiencies
- ✓ Act on a recommendation to resize a virtual machine to a more cost-effective option
- ✓ Verify the action to ensure that the virtual machine was successfully resized

## Prerequisites

Recommendations are available for a variety of scopes and Azure account types. To view the full list of supported account types, see [Understand Cost Management data](#). You must have at least read access to one or more of the following scopes to view cost data. For more information about scopes, see [Understand and work with scopes](#).

- Subscription
- Resource group

If you have a new subscription, you can't immediately use Cost Management features. It might take up to 48 hours before you can use all Cost Management features. Also, you must have active virtual machines with at least 14 days of activity.

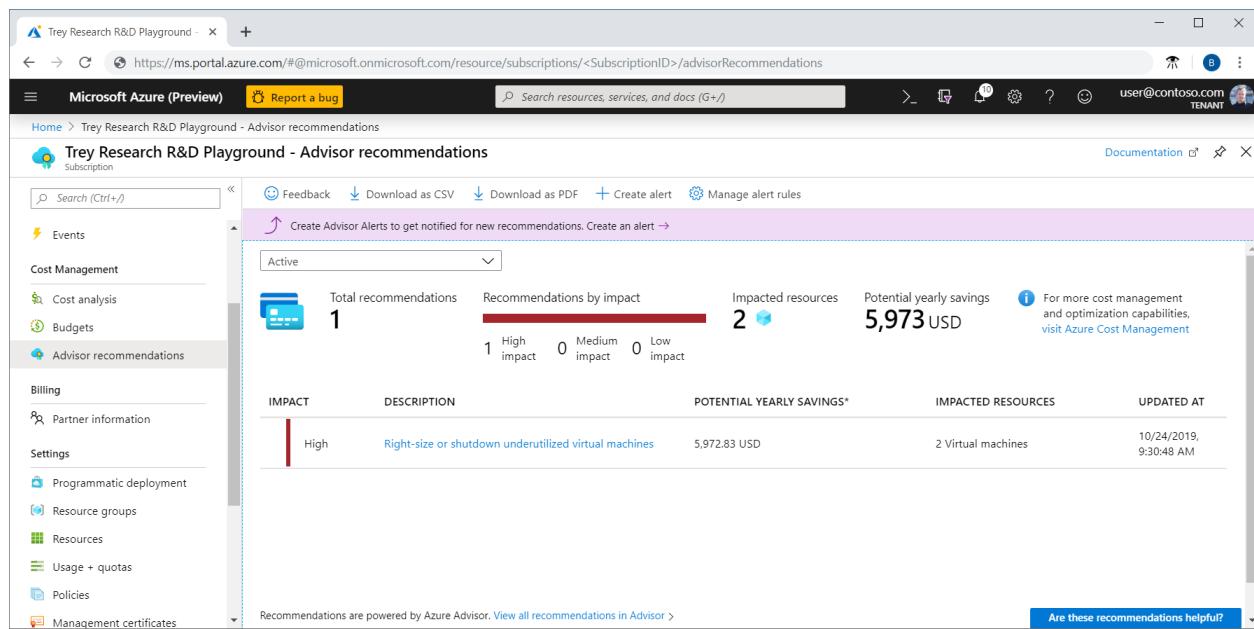
## Sign in to Azure

Sign in to the Azure portal at <https://portal.azure.com>.

# View cost optimization recommendations

To view cost optimization recommendations for a subscription, open the desired scope in the Azure portal and select **Advisor recommendations**.

To view recommendations for a management group, open the desired scope in the Azure portal and select **Cost analysis** in the menu. Use the **Scope** pill to switch to a different scope, such as a management group. Select **Advisor recommendations** in the menu. For more information about scopes, see [Understand and work with scopes](#).



The screenshot shows the Azure portal interface for the Trey Research R&D Playground. The left sidebar is collapsed. The main content area is titled "Trey Research R&D Playground - Advisor recommendations". At the top, there are buttons for "Feedback", "Download as CSV", "Download as PDF", "Create alert", and "Manage alert rules". Below these are sections for "Total recommendations" (1), "Recommendations by impact" (High impact: 1, Medium impact: 0, Low impact: 0), "Impacted resources" (2), and "Potential yearly savings" (\$5,973 USD). A callout notes "For more cost management and optimization capabilities, visit Azure Cost Management". A table below lists the recommendation details: Impact (High), Description (Right-size or shutdown underutilized virtual machines), Potential Yearly Savings\* (5,972.83 USD), Impacted Resources (2 Virtual machines), and Updated At (10/24/2019, 9:30:48 AM). The bottom of the page says "Recommendations are powered by Azure Advisor. View all recommendations in Advisor >" and "Are these recommendations helpful?".

The list of recommendations identifies usage inefficiencies or shows purchase recommendations that can help you save additional money. The totaled **Potential yearly savings** shows the total amount that you can save if you shut down or deallocate all of your VMs that meet recommendation rules. If you don't want to shut them down, you should consider resizing them to a less expensive VM SKU.

The **Impact** category, along with the **Potential yearly savings**, are designed to help identify recommendations that have the potential to save as much as possible.

High impact recommendations include:

- Buy an Azure savings plan to save money on a variety of compute services
- Buy reserved virtual machine instances to save money over pay-as-you-go costs
- Optimize virtual machine spend by resizing or shutting down underutilized instances
- Use Standard Storage to store Managed Disks snapshots

Medium impact recommendations include:

- Reduce costs by eliminating un-provisioned ExpressRoute circuits

- Reduce costs by deleting or reconfiguring idle virtual network gateways

## Act on a recommendation

Azure Advisor monitors your virtual machine usage for seven days and then identifies underutilized virtual machines. Virtual machines whose CPU utilization is five percent or less and network usage is seven MB or less for four or more days are considered low-utilization virtual machines.

The 5% or less CPU utilization setting is the default, but you can adjust the settings. For more information about adjusting the setting, see the [Configure the average CPU utilization rule or the low usage virtual machine recommendation](#).

Although some scenarios can result in low utilization by design, you can often save money by changing the size of your virtual machines to less expensive sizes. Your actual savings might vary if you choose a resize action. Let's walk through an example of resizing a virtual machine.

In the list of recommendations, select the **Right-size or shutdown underutilized virtual machines** recommendation. In the list of virtual machine candidates, choose a virtual machine to resize and then select the virtual machine. The virtual machine's details are shown so that you can verify the utilization metrics. The **potential yearly savings** value is what you can save if you shut down or remove the VM. Resizing a VM will probably save you money, but you won't save the full amount of the potential yearly savings.

**Shut down or resize your virtual machine**

Feedback Download as CSV Download as PDF Create alert Manage alert rules Configure recommendation r...

**Recommendation details**

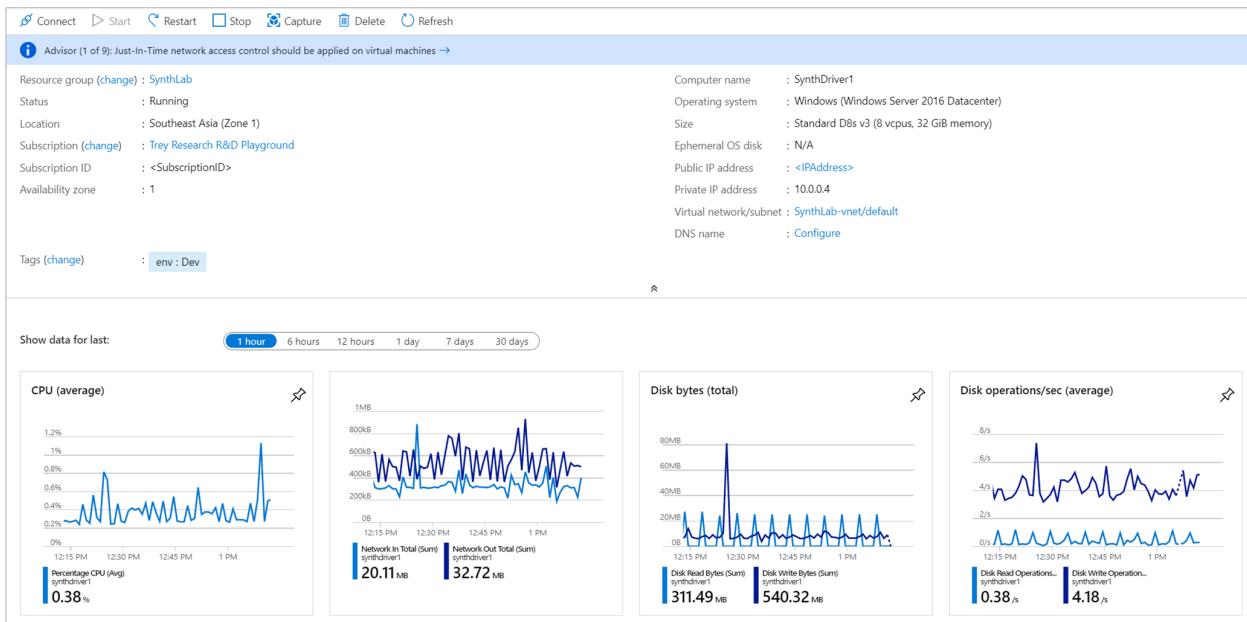
We've analyzed the usage patterns of your virtual machine over the past 7 days and identified virtual machines with low usage. While certain scenarios can result in low utilization by design, you can often save money by managing the size and number of virtual machines. [Learn more](#)

**potential yearly savings\***  
**5,972.83 USD**

**Impacted resources**

Trey Research R&D Playground	No grouping						
Active (2)	<a href="#">Postponed &amp; Dismissed (0)</a>						
<input type="radio"/> Postpone	<input type="radio"/> Dismiss						
SELECT	VIRTUAL MACHINE	RECOMMENDED ACTIONS	POTENTIAL SAVINGS	SUBSCRIPTION	RECOMMENDATION RULE	UPDATED AT	ACTION
<input type="checkbox"/>	 SynthDriver1	<a href="#">Resize Standard_D8s_v3 to Standard_D2s_v3</a> <a href="#">View Usage Patterns</a>	<b>3,348.00 USD</b> (75%)	Trey Research R&D Playground	CPU utilization < 20%	10/24/2019, 9:23:23 AM	<a href="#">Postpone</a>   <a href="#">Dismiss</a>
<input type="checkbox"/>	 testAvi	<a href="#">Resize Standard_DS12_v2 to Standard_DS2_v2</a> <a href="#">View Usage Patterns</a>	<b>2,624.83 USD</b> (63%)	Trey Research R&D Playground	CPU utilization < 20%	10/24/2019, 9:30:48 AM	<a href="#">Postpone</a>   <a href="#">Dismiss</a>

In the VM details, check the utilization of the virtual machine to confirm that it's a suitable resize candidate.



Note the current virtual machine's size. After you've verified that the virtual machine should be resized, close the VM details so that you see the list of virtual machines.

In the list of candidates to shut down or resize, select **Resize** <*FromVirtualMachineSKU*> to <*ToVirtualMachineSKU*>.

This screenshot shows a modal dialog titled 'Select Virtual Machine' with two sections: 'Postpone' and 'Dismiss' at the top. The left section lists two virtual machines:

- SynthDriver1
- testAvi

The right section, titled 'RECOMMENDED ACTIONS', contains two options:

- Resize Standard\_D8s\_v3 to Standard\_D2s\_v3** (highlighted with a red box)
- View Usage Patterns**
- Resize Standard\_DS12\_v2 to Standard\_DS2\_v2**
- View Usage Patterns**

Next, you're presented with a list of available resize options. Choose the one that will give the best performance and cost-effectiveness for your scenario. In the following example, the option chosen resizes from **Standard\_D8s\_v3** to **Standard\_D2s\_v3**.

This screenshot shows the 'VM sizes' selection page. At the top, there's a note: 'If the virtual machine is currently running, changing its size will cause it to be restarted. Stopping the virtual machine may reveal additional sizes. →'. Below this are search and filter options.

The table lists 109 VM sizes, showing columns for VM Size, Offering, Family, vCPUs, RAM, Data disks, Max IOPS, Temporary storage, Premium disk support, and Cost/month (estimated). The row for 'D2s\_v3' is highlighted with a red box.

VM Size	Offering	Family	vCPUs	RAM	Data disks	Max IOPS	Temporary storage	Premium disk support	Cost/month (estimated)
D16s_v3	Standard	General purpose	16	64	32	25600	128	Yes	\$744.00
D2_v2	Standard	General purpose	2	7	8	8x500	100	No	\$117.55
D2_v2	Promo (Exp...)	General purpose	2	7	8	8x500	100	No	\$117.55
D2_v3	Standard	General purpose	2	8	4	4x500	50	No	\$93.00
D2s_v3	Standard	General purpose	2	8	4	3200	16	Yes	\$93.00
D3_v2	Standard	General purpose	4	14	16	16x500	200	No	\$235.10

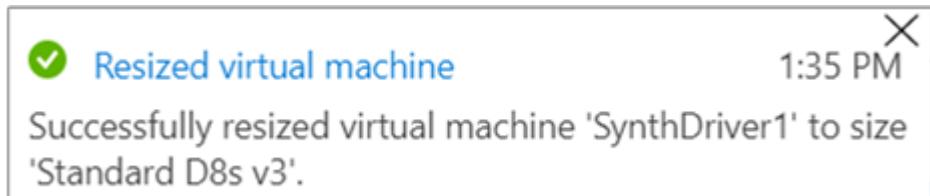
At the bottom, a note states: 'Prices presented are estimates in your local currency that include only Azure infrastructure costs and any discounts for the subscription and location. The prices don't include any applicable software costs. If you purchased Azure services through a reseller, contact your reseller for full pricing details. Final charges will appear in your local currency in cost analysis and billing views.' A 'Resize' button is located at the bottom left.

After you choose a suitable size, select **Resize** to start the resize action.

Resizing requires an actively running virtual machine to restart. If the virtual machine is in a production environment, we recommend that you run the resize operation after business hours. Scheduling the restart can reduce disruptions caused by momentarily unavailability.

## Verify the action

When the VM resizing completes successfully, an Azure notification is shown.



## Next steps

In this tutorial, you learned how to:

- ✓ View cost optimization recommendations to view potential usage inefficiencies
- ✓ Act on a recommendation to resize a virtual machine to a more cost-effective option
- ✓ Verify the action to ensure that the virtual machine was successfully resized

If you haven't already read the Cost Management best practices article, it provides high-level guidance and principles to consider to help manage costs.

[Cost Management best practices](#)

# Reservation recommendations

Article • 03/20/2023

Azure reserved instance (RI) purchase recommendations are provided through Azure Consumption [Reservation Recommendation API](#), [Azure Advisor](#), and through the reservation purchase experience in the Azure portal.

The following steps define how recommendations are calculated:

1. The recommendation engine evaluates the hourly usage for your resources in the given scope over the past 7, 30, and 60 days.
2. Based on the usage data, the engine simulates your costs with and without reservations.
3. The costs are simulated for different quantities, and the quantity that maximizes the savings is recommended.
4. If your resources are shut down regularly, the simulation can't find any savings, and no purchase recommendation is provided.
5. The recommendation calculations include any special discounts that you might have for your on-demand usage rates.

The recommendations account for existing reservations and savings plans. So, previously purchased reservations and savings plans are excluded when providing recommendations.

## Recommendations in the Azure portal

Reservation purchase recommendations are also shown in the Azure portal in the purchase experience. Recommendations are shown with the **Recommended Quantity**. When purchased, the quantity that Azure recommends gives the maximum savings possible. Although you can buy any quantity that you like, if you buy a different quantity your savings aren't optimal.

Let's look at some examples why.

In the following example image for the selected recommendation, Azure recommends a purchase quantity of 6.

Select the product you want to purchase

Reserved VM Instances (RIs) provide a significant discount over pay-as-you-go VM prices by allowing you to pre-purchase the base costs of your VM usage for a period of 1 or 3 years. Reserved instance discount will automatically apply to matching VMs, you don't need to re-deploy resources to get reservation discount. The reservation applies only to hardware usage. Windows is charged separately. [Learn More](#)

Scope \* Shared Billing subscription \* Cost Management Research (1caaa5a3-2b66-438e-8...)

**Recommended** All Products

Showing recommendations based on your usage over the last 60 d... [Learn more](#)

Name	Region	Instance flexibility group	vCPUs	RAM (GB)	Term	Billing frequency	Recommended Quantity
Standard_DS1_v2	West US 2	DSv2 Series	1	3.5	Three Years	Monthly	16 - See details
Standard_DS1_v2	East US 2	DSv2 Series	1	3.5	Three Years	Monthly	14 - See details
Standard_DS3_v2	East US	DSv2 Series	4	14	Three Years	Monthly	9 - See details
Standard_DS2_v2	Southeast Asia	DSv2 Series	2	7	Three Years	Monthly	8 - See details
Standard_DS2_v2	Central US	DSv2 Series	2	7	Three Years	Monthly	6 - See details
Standard_DS2_v2	East US	DSv2 Series	2	7	Three Years	Monthly	5 - See details
Standard_F2s_v2	Southeast Asia	FSv2 Series	2	4	Three Years	Monthly	5 - See details

Not seeing what you want? [Browse all products](#).

[Select](#) [Close](#)

More information about the recommendation appears when you select **See details**. The following image shows details about the recommendation. The quantity recommended is calculated for the highest possible usage and it's based on your historical usage. Your recommendation might not be for 100% utilization if you have inconsistent usage. In the example, notice that utilization fluctuated over time. The cost of the reservation, possible savings, and utilization percentage is shown.

Select the product you want to purchase

Calculate costs

Reservation quantity (6 recommended)	7,642	Cost of reservation (US\$)	0	New estimated total (USD)	7,642	Savings (USD)	11,383	Utilization	83%
6	1,274 per reservation	Remaining	Was 19,025						

Analyze data

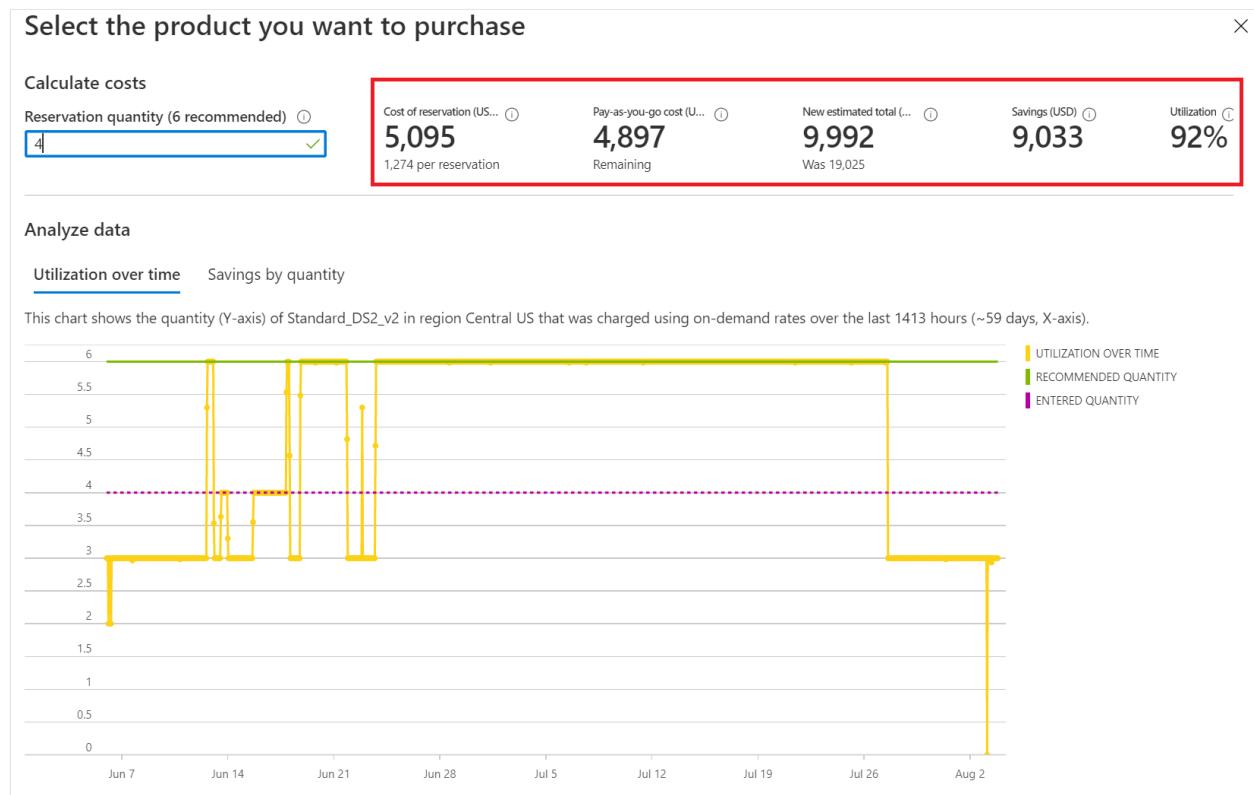
Utilization over time Savings by quantity

This chart shows the quantity (Y-axis) of Standard\_DS2\_v2 in region Central US that was charged using on-demand rates over the last 1413 hours (~59 days, X-axis).

Add to cart Back

The chart and estimated values change when you increase the recommended quantity. When you increase the reservation quantity, your savings are reduced because you end up with reduced reservation use. In other words, you pay for reservations that aren't fully used.

If you lower the reservation quantity, your savings are also reduced. Although utilization is increased, there might be periods when your reservations don't fully cover your use. Usage beyond your reservation quantity is used by more expensive pay-as-you-go resources. The following example image illustrates the point. We've manually reduced the reservation quantity to 4. The reservation utilization is increased, but the overall savings are reduced because pay-as-you go costs are present.



To maximize savings with reservations, try to purchase reservations as close to the recommendation as possible.

## Recommendations in Azure Advisor

Reservation purchase recommendations are available in Azure Advisor. Keep in mind the following points:

- Advisor has only single-subscription scope recommendations. If you want to see recommendations for the entire billing scope (Billing account or billing profile), then:
  - In the Azure portal, navigate to **Reservations > Add** and then select the type that you want to see the recommendations for.

- Recommendations available in Advisor consider your past 30-day usage trend.
- The recommendations quantity and savings are for a three-year reservation, where available. If a three-year reservation isn't sold for the service, the recommendation is calculated using the one-year reservation price.
- The recommendation calculations include any special discounts that you might have on your on-demand usage rates.
- If you purchase a shared-scope reservation, Advisor reservation purchase recommendations can take up to five days to disappear.
- Azure classic compute resources such as classic VMs are explicitly excluded from reservation recommendations. Microsoft recommends that users avoid making long-term commitments to legacy services that are being deprecated.

## Other expected API behavior

- When using a look-back period of seven days, you might not get recommendations when VMs are shut down for more than a day.

## Next steps

- Get [Reservation recommendations using REST APIs](#).
- Learn about [how the Azure reservation discount is applied to virtual machines](#).

# Determine what reservation to purchase

Article • 06/27/2023

All reservations, except Azure Databricks, are applied on an hourly basis. You should purchase reservations based on consistent base usage. There are multiple ways to determine what to purchase and this article helps you determine which reservation you should purchase.

Purchasing more capacity than your historical usage results in an underutilized reservation. You should avoid underutilization whenever possible. Unused reserved capacity doesn't carry over from one hour to next. Usage exceeding the reserved quantity is charged using more expensive pay-as-you-go rates.

## Analyze usage data

Use the following sections to help analyze your daily usage data to determine your baseline usage and what reservation to purchase. Before you begin, review the [View and download your Azure usage and charges](#) article to get details about how to download the usage file.

### Analyze usage for a VM reserved instance purchase

Identify the right VM size for your purchase. For example, a reservation purchased for ES series VMs doesn't apply to E series VMs, and vice-versa.

Promo series VMs don't get a reservation discount, so remove them from your analysis.

To narrow down to eligible VM usage, apply the following filters on your usage data:

- Filter `MeterCategory` to `Virtual Machines`.
- Get `ServiceType` information from `AdditionalInfo`. The information suggests the right VM size. For example, `D2s_v3`.
- Use the `ResourceLocation` field to determine the usage data center.

Ignore resources that have less than 24 hours of usage in a day.

Here's an example of the usage file showing the usage file with filters applied. In the example, `AdditionalInfo` suggests a `D2s_v3` virtual machine.

M	N	O	P	Q	R	S	T	U	V	
Number	MeterId	ServiceFamily	MeterCategory	MeterSubCategory	MeterRegion	MeterName	BillingCurrency	ResourceLocation	ConsumedService	AdditionalInfo
11 -68652	2c57ed84-Compute	Virtual Machines	BS Series	Virginia	B2s	USD	EastUS	Microsoft.Capacity		
12 -87169	Se274480-Compute	Virtual Machines	BS Series	Netherlands	B1s	USD	westeurope	Microsoft.Capacity		
13 -87169	Se274480-Compute	Virtual Machines	BS Series	Netherlands	B1s	USD	westeurope	Microsoft.Capacity		
14 -87169	Se274480-Compute	Virtual Machines	BS Series	Netherlands	B1s	USD	westeurope	Microsoft.Capacity		
15 -01916	9a721778-Compute	Virtual Machines	D/DS Series	California	D1/D1s	USD	WestUS	Microsoft.Capacity		
16 -87169	Se274480-Compute	Virtual Machines	BS Series	Netherlands	B1s	USD	westeurope	Microsoft.Capacity		
17 -44977	ae331802-Compute	Virtual Machines	Dv3/Dsv3 Series	Virginia	D2 v3/D2s v3	USD	EastUS	Microsoft.Capacity		
18 -87169	Se274480-Compute	Virtual Machines	BS Series	Netherlands	B1s	USD	westeurope	Microsoft.Capacity		
82 -44977	ae331802-Compute	Virtual Machines	Dv3/Dsv3 Series	Virginia	D2 v3/D2s v3	USD	EastUS	Microsoft.Compute	{"UsageType": "ComputeHR", "ImageType": "Canonical", "ServiceType": "Standard_D2s_v3", "VMName": "VMName"}	
763 -44977	ae331802-Compute	Virtual Machines	Dv3/Dsv3 Series	Virginia	D2 v3/D2s v3	USD	EastUS	Microsoft.Compute	{"UsageType": "ComputeHR", "ImageType": "Canonical", "ServiceType": "Standard_D2s_v3", "VMName": "VMName"}	
866 -44977	ae331802-Compute	Virtual Machines	Dv3/Dsv3 Series	Virginia	D2 v3/D2s v3	USD	EastUS	Microsoft.Compute	{"UsageType": "ComputeHR", "ImageType": "Canonical", "ServiceType": "Standard_D2s_v3", "VMName": "VMName"}	
867 -44977	ae331802-Compute	Virtual Machines	Dv3/Dsv3 Series	Virginia	D2 v3/D2s v3	USD	EastUS	Microsoft.Compute	{"UsageType": "ComputeHR", "ImageType": "Canonical", "ServiceType": "Standard_D2s_v3", "VMName": "VMName"}	

If you want to analyze at the instance size family level, you can get the instance size flexibility values from <https://isfratio.blob.core.windows.net/isfratio/ISFRatio.csv>. Combine the values with your data to do the analysis. For more information about instance size flexibility, see [Virtual machine size flexibility with Reserved VM Instances](#).

## Analyze usage for an Azure Synapse Analytics reserved instance purchase

Reserved capacity applies to Azure Synapse Analytics DWU pricing. It doesn't apply to Azure Synapse Analytics license cost or any costs other than compute.

To narrow eligible usage, apply the following filters to your usage data:

- Filter MeterCategory for **SQL Database**.
- Filter MeterName for **vCore**.
- Filter MeterSubCategory for all usage records that have *Compute* in the name.

From **AdditionalInfo**, get the **vCores** value. It tells you how many vCores were used. The quantity is **vCores** multiplied by the number of hours the database was used.

The data informs you about the consistent usage for:

- Combination of database type. For example, managed instance or elastic pool per single database.
- Service tier. For example, general purpose or business critical.
- Generation. For example, Gen 5.
- Resource Location

## Analysis for Azure Synapse Analytics

Reserved capacity applies to Azure Synapse Analytics DWU usage and is purchased in increments on 100 DWU. To narrow eligible usage, apply the following filters on your usage data:

- Filter MeterName for **100 DWUs**.
- Filter Meter Sub-Category for **Compute Optimized Gen2**.

Use the **Resource Location** field to determine the usage for Azure Synapse Analytics in a region.

Azure Synapse Analytics usage can scale up and down throughout the day. Talk to the team that managed the Azure Synapse Analytics instance to learn about the base usage.

Go to Reservations in the Azure portal and purchase Azure Synapse Analytics reserved capacity in multiples of 100 DWUs.

## Reservation purchase recommendations

Reservation purchase recommendations are calculated by analyzing your hourly usage data over last 7, 30, and 60 days. Azure calculates what your costs would have been if you had a reservation and compares it with your actual pay-as-you-go costs incurred over the time duration. The calculation is performed for every quantity that you used during the time frame. The quantity that maximizes your savings is recommended.

For example, you might use 500 VMs most of the time, but sometimes usage spikes to 700 VMs. In this example, Azure calculates your savings for both the 500 and 700 VM quantities. Since the 700 VM usage is sporadic, the recommendation calculation determines that savings are maximized for a 500 VM reservation purchase and the recommendation is provided for the 500 quantity.

Note the following points:

- Reservation recommendations are calculated using the on-demand usage rates that apply to you.
- Recommendations are calculated for individual sizes, not for the instance size family.
- The recommended quantity for a scope is reduced on the same day that you purchase reservations for the scope.
  - However, an update for the reservation quantity recommendation across scopes can take up to 25 days. For example, if you purchase based on shared scope recommendations, the single subscription scope recommendations can take up to 25 days to adjust down.
- Currently, Azure doesn't generate recommendations for the management group scope.

## Recommendations in the Azure portal

Reservation purchases calculated by the recommendations engine are shown on the **Recommended** tab in the [Azure portal](#). Here's an example image.

The screenshot shows the Azure portal interface for selecting products to purchase. On the left, there's a sidebar with 'Purchase reservations' and a list of services like Virtual machine, Azure Blob Storage, etc. The main area is titled 'Select the product you want to purchase' and contains a table of recommendations. The table columns include Name, Region, Instance flexibility group, vCPUs, RAM (GB), Term, Billing frequency, and Recommended Quantity. One row for 'Standard\_DS2\_v2' in Central US is selected and highlighted with a red border.

Name	Region	Instance flexibility group	vCPUs	RAM (GB)	Term	Billing frequency	Recommended Quantity
Standard_DS1_v2	West US 2	DSv2 Series	1	3.5	Three Years	Monthly	16 - See details
Standard_DS1_v2	East US 2	DSv2 Series	1	3.5	Three Years	Monthly	14 - See details
Standard_DS3_v2	East US	DSv2 Series	4	14	Three Years	Monthly	9 - See details
Standard_DS2_v2	Southeast Asia	DSv2 Series	2	7	Three Years	Monthly	8 - See details
Standard_DS2_v2	Central US	DSv2 Series	2	7	Three Years	Monthly	6 - See details
Standard_DS2_v2	East US	DSv2 Series	2	7	Three Years	Monthly	5 - See details
Standard_F2s_v2	Southeast Asia	FSv2 Series	2	4	Three Years	Monthly	5 - See details

Learn more about [recommendations](#).

## Recommendations in the Cost Management Power BI app

Enterprise Agreement customers can use the VM RI Coverage reports for VMs and purchase recommendations. The coverage reports show total usage and the usage that's covered by reserved instances.

1. Get the [Cost Management App](#).
2. Go to the VM RI Coverage report – Shared or Single scope, depending on which scope you want to purchase at.
3. Select the region, instance size family to see the usage, RI coverage, and the purchase recommendation for the selected filter.

## Recommendations in Azure Advisor

Reservation purchase recommendations are available in [Azure Advisor](#).

- Advisor has only single-subscription scope recommendations. If you want to see recommendations for the entire billing scope (Billing account or billing profile), then:
- In the Azure portal, navigate to Reservations > Add and then select the type that you want to see the recommendations for.

- The recommendations quantity and savings are for a three-year reservation, where available. If a three-year reservation isn't sold for the service, the recommendation is calculated using the one-year reservation price.
- The recommendation calculations include any special discounts that you might have on your on-demand usage rates.
- If you purchase a shared-scope reservation, Advisor reservation purchase recommendations can take up to five days to disappear.
- Azure classic compute resources such as classic VMs are explicitly excluded from reservation recommendations. Microsoft recommends that users avoid making long-term commitments to legacy services that are being deprecated.

## Recommendations using APIs

Use the [Reservation Recommendations](#) REST API to view recommendations programmatically.

## Next steps

- [Manage Azure Reservations](#)
- [Understand reservation usage for your subscription with pay-as-you-go rates](#)
- [Understand reservation usage for your Enterprise enrollment](#)
- [Windows software costs not included with reservations](#)

# Buy a reservation

Article • 12/07/2022

Azure Reservations help you save money by committing to one-year or three-years plans for many Azure resources. Before you enter a commitment to buy a reservation, be sure to review the following sections to prepare for your purchase.

## Who can buy a reservation

To buy a reservation, you must have owner role or reservation purchaser role on an Azure subscription that's of type Enterprise (MS-AZR-0017P or MS-AZR-0148P) or Pay-As-You-Go (MS-AZR-0003P or MS-AZR-0023P) or Microsoft Customer Agreement.

Cloud solution providers can use the Azure portal or [Partner Center](#) to purchase Azure Reservations. You will not be able to purchase a reservation if you have a custom role that mimics owner role or reservation purchaser role on an Azure subscription, you must use built-in owner or built-in reservation purchaser role.

Enterprise Agreement (EA) customers can limit purchases to EA admins by disabling the **Add Reserved Instances** option in the EA Portal. Direct EA customers can now disable Reserved Instance setting in [Azure portal](#). Navigate to Policies menu to change settings.

EA admins must have owner or reservation purchaser access on at least one EA subscription to purchase a reservation. The option is useful for enterprises that want a centralized team to purchase reservations.

A reservation discount only applies to resources associated with subscriptions purchased through Enterprise, Cloud Solution Provider (CSP), Microsoft Customer Agreement and individual plans with pay-as-you-go rates.

## Scope reservations

You can scope a reservation to a subscription or resource groups. Setting the scope for a reservation selects where the reservation savings apply. When you scope the reservation to a resource group, reservation discounts apply only to the resource group—not the entire subscription.

## Reservation scoping options

You have four options to scope a reservation, depending on your needs:

- **Single resource group scope** — Applies the reservation discount to the matching resources in the selected resource group only.
- **Single subscription scope** — Applies the reservation discount to the matching resources in the selected subscription.
- **Shared scope** — Applies the reservation discount to matching resources in eligible subscriptions that are in the billing context. If a subscription was moved to different billing context, the benefit will no longer be applied to this subscription and will continue to apply to other subscriptions in the billing context.
  - For Enterprise Agreement customers, the billing context is the enrollment. The reservation shared scope would include multiple Active Directory tenants in an enrollment.
  - For Microsoft Customer Agreement customers, the billing scope is the billing profile.
  - For individual subscriptions with pay-as-you-go rates, the billing scope is all eligible subscriptions created by the account administrator.
- **Management group** — Applies the reservation discount to the matching resource in the list of subscriptions that are a part of both the management group and billing scope. To buy a reservation for a management group, you must have at least read permission on the management group and be a reservation owner or reservation purchaser on the billing subscription.

While applying reservation discounts on your usage, Azure processes the reservation in the following order:

1. Reservations with a single resource group scope
2. Reservations with a single subscription scope
3. Reservations scoped to a management group
4. Reservations with a shared scope (multiple subscriptions), described previously

You can always update the scope after you buy a reservation. To do so, go to the reservation, select **Configuration**, and rescope the reservation. Rescoping a reservation isn't a commercial transaction. Your reservation term isn't changed. For more information about updating the scope, see [Update the scope after you purchase a reservation](#).

Home > Reservations > 11111111-1111-1111-1111-111111111111/22222222-2222-2222-2222-222222222222 | Configuration

Reservation | Directory: Unknown directory

Search (Ctrl+ /) < Save Discard View FAQs

Overview Access control (IAM)

Settings Configuration

Renewal Properties

Support + troubleshooting New Support Request

Scope

The reservation's scope can cover one subscription, one resource group or multiple subscriptions. If you select **Shared** - The reservation discount is applied to the matching resources in eligible subscriptions within your billing context. For enterprise customers, the billing context is the enrollment. For Pay-As-You-Go customers, the billing context is all eligible subscriptions created by the account administrator. For Microsoft Customer Agreement, the billing context is the billing profile.

**Management group** - The reservation discount is applied to the matching resources in the list of subscriptions that are a part of both management group and billing scope.

**Single subscription** - The reservation discount is applied to the matching resources in the subscription you select.

**Single resource group** - The reservation discount is applied only to the matching resources in the resource group you select. If you move or delete the resource group, the reservation discount will not apply.

Scope \* Management Group (preview)

Management Group \* Select a management group

# Discounted subscription and offer types

Reservation discounts apply to the following eligible subscriptions and offer types.

- Enterprise agreement (offer numbers: MS-AZR-0017P or MS-AZR-0148P)
- Microsoft Customer Agreement subscriptions.
- Individual plans with pay-as-you-go rates (offer numbers: MS-AZR-0003P or MS-AZR-0023P)
- CSP subscriptions

Resources that run in a subscription with other offer types don't receive the reservation discount.

# Purchase reservations

You can purchase reservations from Azure portal, APIs, PowerShell, CLI. Read the following articles that apply to you when you're ready to make a reservation purchase:

- [App Service](#)
- [App Service - JBoss EA Integrated Support](#)
- [Azure Backup](#)
- [Azure Cache for Redis](#)
- [Azure Data Factory](#)
- [Azure Database for MariaDB](#)
- [Azure Database for MySQL](#)
- [Azure Database for PostgreSQL](#)
- [Azure Blob storage](#)
- [Azure Files](#)
- [Azure VMware Solution](#)

- [Azure Cosmos DB](#)
- [Azure SQL Edge](#)
- [Databricks](#)
- [Data Explorer](#)
- [Dedicated Host](#)
- [Disk Storage](#)
- [SAP HANA Large Instances](#)
- [Software plans](#)
- [SQL Database](#)
- [Synapse Analytics - data warehouse](#)
- [Synapse Analytics - Pre-purchase](#)
- [Virtual machines](#)
- [Virtual machine software](#)

## Buy reservations with monthly payments

You can pay for reservations with monthly payments. Unlike an up-front purchase where you pay the full amount, the monthly payment option divides the total cost of the reservation evenly over each month of the term. The total cost of up-front and monthly reservations is the same and you don't pay any extra fees when you choose to pay monthly.

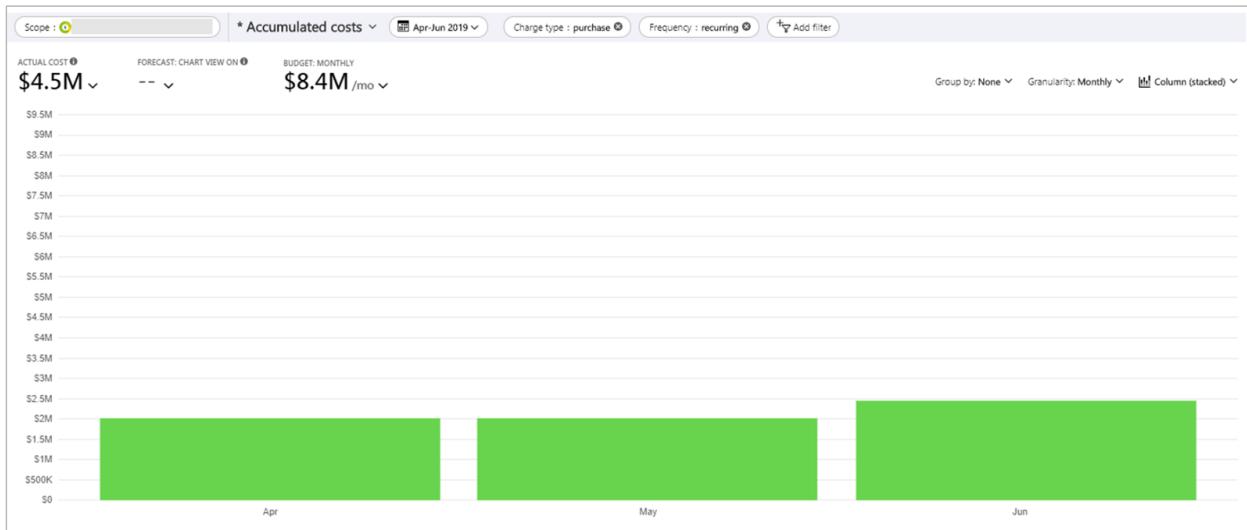
If reservation is purchased using Microsoft customer agreement (MCA), your monthly payment amount may vary, depending on the current month's market exchange rate for your local currency.

Monthly payments are not available for: Databricks, Synapse Analytics - Pre-purchase, SUSE Linux reservations, Red Hat Plans and Azure Red Hat OpenShift Licenses.

## View payments made

You can view payments that were made using APIs, usage data, and in cost analysis. For reservations paid for monthly, the frequency value is shown as **recurring** in usage data and Reservation Charges API. For reservations paid up front, the value is shown as **onetime**.

Cost analysis shows monthly purchases in the default view. Apply the **purchase** filter to **Charge type** and **recurring** for **Frequency** to see all purchases. To view only reservations, apply a filter for **Reservation**.



## Exchange and refunds

Like other reservations, you can refund or exchange reservations purchased with monthly billing.

When you exchange a reservation that's paid for monthly, the total lifetime cost of the new purchase should be greater than the leftover payments that are canceled for the returned reservation. There are no other limits or fees for exchanges. You can exchange a reservation that's paid for up front to purchase a new reservation that's billed monthly. However, the lifetime value of the new reservation should be greater than the prorated value of the reservation being returned.

If you cancel a reservation that's paid for monthly, canceled future payments accrue towards the \$50,000 USD refund limit.

For more information about exchange and refunds, see [Self-service exchanges and refunds for Azure Reservations](#).

## Reservation notifications

Depending on how you pay for your Azure subscription, email reservation notifications are sent to the following users in your organization. Notifications are sent for various events including:

- Purchase
- Upcoming reservation expiration
- Expiry
- Renewal
- Cancellation
- Scope change

Notifications are sent to the following users:

- Customers with EA subscriptions
  - Notifications are sent to the EA notification contacts, EA admin, reservation owners, and the reservation administrator.
- Customers with Microsoft Customer Agreement (Azure Plan)
  - Notifications are sent to the reservation owners and the reservation administrator.
- Cloud Solution Provider and new commerce partners
  - Emails are sent to the partner notification contact.
- Individual subscription customers with pay-as-you-go rates
  - Emails are sent to users who are set up as account administrators, reservation owners, and the reservation administrator.

## Next steps

- [Learn more about Reservations permissions](#)
- [Manage Reservations for Azure resources](#)
- [Automate using REST APIs](#)
- [Automate using Azure PowerShell](#)
- [Automate using CLI](#)

# Permissions to view and manage Azure reservations

Article • 03/03/2023

This article explains how reservation permissions work and how users can view and manage Azure reservations in the Azure portal and with Azure PowerShell.

## ⓘ Note

We recommend that you use the Azure Az PowerShell module to interact with Azure. See [Install Azure PowerShell](#) to get started. To learn how to migrate to the Az PowerShell module, see [Migrate Azure PowerShell from AzureRM to Az](#).

## Who can manage a reservation by default

By default, the following users can view and manage reservations:

- The person who buys a reservation and the account administrator of the billing subscription used to buy the reservation are added to the reservation order.
- Enterprise Agreement and Microsoft Customer Agreement billing administrators.
- Users with elevated access to manage all Azure subscriptions and management groups
- A Reservation administrator for reservations in their Azure Active Directory (Azure AD) tenant (directory)
- A Reservation reader has read-only access to reservations in their Azure Active Directory tenant (directory)

The reservation lifecycle is independent of an Azure subscription, so the reservation isn't a resource under the Azure subscription. Instead, it's a tenant-level resource with its own Azure RBAC permission separate from subscriptions. Reservations don't inherit permissions from subscriptions after the purchase.

## View and manage reservations

If you're a billing administrator, use following steps to view and manage all reservations and reservation transactions in the Azure portal.

1. Sign into the [Azure portal](#) and navigate to **Cost Management + Billing**.

- If you're an EA admin, in the left menu, select **Billing scopes** and then in the list of billing scopes, select one.
- If you're a Microsoft Customer Agreement billing profile owner, in the left menu, select **Billing profiles**. In the list of billing profiles, select one.

2. In the left menu, select **Products + services > Reservations**.
3. The complete list of reservations for your EA enrollment or billing profile is shown.
4. Billing administrators can take ownership of a reservation by selecting one or multiple reservations, selecting **Grant access** and selecting **Grant access** in the window that appears. For a Microsoft Customer Agreement, user should be in the same Azure Active Directory (Azure AD) tenant (directory) as the reservation.

## Add billing administrators

Add a user as billing administrator to an Enterprise Agreement or a Microsoft Customer Agreement in the Azure portal.

- For an Enterprise Agreement, add users with the *Enterprise Administrator* role to view and manage all reservation orders that apply to the Enterprise Agreement. Enterprise administrators can view and manage reservations in **Cost Management + Billing**.
  - Users with the *Enterprise Administrator (read only)* role can only view the reservation from **Cost Management + Billing**.
  - Department admins and account owners can't view reservations *unless* they're explicitly added to them using Access control (IAM). For more information, see [Managing Azure Enterprise roles](#).
- For a Microsoft Customer Agreement, users with the billing profile owner role or the billing profile contributor role can manage all reservation purchases made using the billing profile. Billing profile readers and invoice managers can view all reservations that are paid for with the billing profile. However, they can't make changes to reservations. For more information, see [Billing profile roles and tasks](#).

## View reservations with Azure RBAC access

If you purchased the reservation or you're added to a reservation, use the following steps to view and manage reservations in the Azure portal.

1. Sign in to the [Azure portal](#).
2. Select **All Services > Reservations** to list reservations that you have access to.

# Manage subscriptions and management groups with elevated access

You can elevate a user's [access to manage all Azure subscriptions and management groups](#).

After you have elevated access:

1. Navigate to All Services > Reservation to see all reservations that are in the tenant.
2. To make modifications to the reservation, add yourself as an owner of the reservation order using Access control (IAM).

## Grant access to individual reservations

Users who have owner access on the reservations and billing administrators can delegate access management for an individual reservation order in the Azure portal.

To allow other people to manage reservations, you have two options:

- Delegate access management for an individual reservation order by assigning the Owner role to a user at the resource scope of the reservation order. If you want to give limited access, select a different role.  
For detailed steps, see [Assign Azure roles using the Azure portal](#).
- Add a user as billing administrator to an Enterprise Agreement or a Microsoft Customer Agreement:
  - For an Enterprise Agreement, add users with the *Enterprise Administrator* role to view and manage all reservation orders that apply to the Enterprise Agreement. Users with the *Enterprise Administrator (read only)* role can only view the reservation. Department admins and account owners can't view reservations unless they're explicitly added to them using Access control (IAM). For more information, see [Managing Azure Enterprise roles](#).

*Enterprise Administrators can take ownership of a reservation order and they can add other users to a reservation using Access control (IAM).*

- For a Microsoft Customer Agreement, users with the billing profile owner role or the billing profile contributor role can manage all reservation purchases made using the billing profile. Billing profile readers and invoice managers can view all reservations that are paid for with the billing profile. However, they can't make

changes to reservations. For more information, see [Billing profile roles and tasks](#).

## Grant access with PowerShell

Users that have owner access for reservations orders, users with elevated access, and [User Access Administrators](#) can delegate access management for all reservation orders they have access to.

Access granted using PowerShell isn't shown in the Azure portal. Instead, you use the `get-AzRoleAssignment` command in the following section to view assigned roles.

## Assign the owner role for all reservations

Use the following Azure PowerShell script to give a user Azure RBAC access to all reservations orders in their Azure AD tenant (directory).

Azure PowerShell

```
Import-Module Az.Accounts
Import-Module Az.Resources

Connect-AzAccount -Tenant <TenantId>

$response = Invoke-AzRestMethod -Path
/providers/Microsoft.Capacity/reservations?api-version=2020-06-01 -Method
GET

$responseJSON = $response.Content | ConvertFrom-JSON

$reservationObjects = $responseJSON.value

foreach ($reservation in $reservationObjects)
{
 $reservationOrderId = $reservation.id.substring(0, 84)
 Write-Host "Assigning Owner role assignment to "$reservationOrderId
 New-AzRoleAssignment -Scope $reservationOrderId -ObjectId <ObjectId> -
 RoleDefinitionName Owner
}
```

When you use the PowerShell script to assign the ownership role and it runs successfully, a success message isn't returned.

## Parameters

**-ObjectId** Azure AD ObjectId of the user, group, or service principal.

- Type: String
- Aliases: Id, PrincipalId
- Position: Named
- Default value: None
- Accept pipeline input: True
- Accept wildcard characters: False

**-TenantId** Tenant unique identifier.

- Type: String
- Position: 5
- Default value: None
- Accept pipeline input: False
- Accept wildcard characters: False

## Tenant-level access

User Access Administrator rights are required before you can grant users or groups the Reservations Administrator and Reservations Reader roles at the tenant level. In order to get User Access Administrator rights at the tenant level, follow [Elevate access](#) steps.

## Add a Reservations Administrator role or Reservations Reader role at the tenant level

You can assign these roles from [Azure portal](#).

1. Sign in to the Azure portal and navigate to **Reservations**.
2. Select a reservation that you have access to.
3. At the top of the page, select **Role Assignment**.
4. Select the **Roles** tab.
5. To make modifications, add a user as a Reservations Administrator or Reservations Reader using Access control.

## Add a Reservation Administrator role at the tenant level using Azure PowerShell script

Use the following Azure PowerShell script to add a Reservation Administrator role at the tenant level with PowerShell.

## Azure PowerShell

```
Import-Module Az.Accounts
Import-Module Az.Resources
Connect-AzAccount -Tenant <TenantId>
New-AzRoleAssignment -Scope "/providers/Microsoft.Capacity" -PrincipalId
<ObjectId> -RoleDefinitionName "Reservations Administrator"
```

## Parameters

-ObjectId Azure AD ObjectId of the user, group, or service principal.

- Type: String
- Aliases: Id, PrincipalId
- Position: Named
- Default value: None
- Accept pipeline input: True
- Accept wildcard characters: False

-TenantId Tenant unique identifier.

- Type: String
- Position: 5
- Default value: None
- Accept pipeline input: False
- Accept wildcard characters: False

## Assign a Reservation Reader role at the tenant level using Azure PowerShell script

Use the following Azure PowerShell script to assign the Reservation Reader role at the tenant level with PowerShell.

## Azure PowerShell

```
Import-Module Az.Accounts
Import-Module Az.Resources

Connect-AzAccount -Tenant <TenantId>

New-AzRoleAssignment -Scope "/providers/Microsoft.Capacity" -PrincipalId
<ObjectId> -RoleDefinitionName "Reservations Reader"
```

## Parameters

**-ObjectId** Azure AD ObjectId of the user, group, or service principal.

- Type: String
- Aliases: Id, PrincipalId
- Position: Named
- Default value: None
- Accept pipeline input: True
- Accept wildcard characters: False

**-TenantId** Tenant unique identifier.

- Type: String
- Position: 5
- Default value: None
- Accept pipeline input: False
- Accept wildcard characters: False

## Next steps

- [Manage Azure Reservations](#).

# View Azure reservations as a Cloud Solution Provider (CSP)

Article • 12/07/2022

Cloud Solution Providers can access reservations that are purchased for their customers. Use the following information to view reservations in the Azure portal.

1. Contact your global admin to get yourself added as an **admin agent** in your tenant. The option is available to global admins in the Partner Center. It's under **Settings** (the gear symbol on the top right of the page) > **User management**.
2. After you have admin agent privilege, go to the Azure portal using the **Admin on Behalf Of** link.
3. Navigate to Partner Center > **Customers** > expand customer details > **Microsoft Azure Management Portal**.
4. In the Azure portal, go to **Reservations**.

## Note

Being a guest in the customer's tenant prevents you from viewing reservations. If you have guest access, you need to remove it from the tenant. Admin agent privilege doesn't override guest access.

- To remove your guest access in the Partner Center, navigate to **My Account** > [Organizations](#) and then select **Leave organization**.

Alternately, ask another user who can access the reservation to add your guest account to the reservation order.

## Next steps

- [View Azure reservations](#)

# Self-service exchanges and refunds for Azure Reservations

Article • 05/03/2023

Azure Reservations provide flexibility to help meet your evolving needs. Reservation products are interchangeable with each other if they're the same type of reservation. For example, you can exchange multiple compute reservations including Azure Dedicated Host, Azure VMware Solution, and Azure Virtual Machines with each other all at once. You can also exchange multiple SQL database reservation types including SQL Managed Instances and Elastic Pool with each other.

However, you can't exchange dissimilar reservations. For example, you can't exchange an Azure Cosmos DB reservation for SQL Database.

You can also exchange a reservation to purchase another reservation of a similar type in a different region. For example, you can exchange a reservation that's in West US 2 region for one that's in West Europe region.

## ⓘ Note

Exchanges will be unavailable for all compute reservations - Azure Reserved Virtual Machine Instances, Azure Dedicated Host reservations, and Azure App Services reservations - purchased on or after **January 1, 2024**. Compute reservations purchased **prior to January 1, 2024** will reserve the right to **exchange one more time** after the policy change goes into effect. Microsoft launched Azure savings plan for compute and it's designed to help you save broadly on predictable compute usage. The savings plan provides more flexibility needed to accommodate changes such as virtual machine series and regions. With savings plan providing the flexibility automatically, we're adjusting our reservations exchange policy. You can continue to use instance size flexibility for VM sizes, but we'll no longer support exchanging instance series or regions for Azure Reserved Virtual Machine Instances, Azure Dedicated Host reservations, and Azure App Services reservations. For more information about the exchange policy change, see [Changes to the Azure reservation exchange policy](#).

You may **trade-in** your Azure compute reservations for a savings plan. Or, you may continue to use and purchase reservations for those predictable, stable workloads where you know the specific configuration you'll need and want additional savings. Learn more about [Azure savings plan for compute and how it works with reservations](#).

When you exchange a reservation, you can change your term from one-year to three-year. Or, you can change the term from three-year to one-year.

You can also refund reservations, but the sum total of all canceled reservation commitment in your billing scope (such as EA, Microsoft Customer Agreement, and Microsoft Partner Agreement) can't exceed USD 50,000 in a 12 month rolling window.

The following reservations aren't eligible for refunds:

- Azure Databricks reserved capacity
- Synapse Analytics Pre-purchase plan
- Azure VMware solution by CloudSimple
- Azure Red Hat Open Shift
- Red Hat plans
- SUSE Linux plans

 **Note**

- You must have owner or Reservation administrator access on the Reservation Order to exchange or refund an existing reservation. You can Add or change users who can manage a reservation.
- Microsoft is not currently charging early termination fees for reservation refunds. We might charge the fees for refunds made in the future. We currently don't have a date for enabling the fee.

## How to exchange or refund an existing reservation

You can exchange your reservation from [Azure portal](#).

## 1. Select the reservations that you want to refund and select Exchange.

Home > Reservations > Exchange

**Exchange**

Return Purchase Review

Configure the reservations that you want to return. These reservations will be canceled immediately and the original charges will be reversed. It can take up to 7 business days for the charges to be refunded. Learn more about how refund and exchange are processed.

NAME	PRODUCT NAME	REGION	QUANTITY	PRORATED COST (PER UNIT)	QUANTITY TO RETURN	REFUND SUBTOTAL
VirtualMachines_Reservation_03-26-2019_07-07	Standard_B1s	South Central US	1	<Prorated cost>	1	<Prorated item total>
virtualMachines_Reservation_03-28-2019_08-55	Standard_B1s	South Central US	1	<Prorated cost>	1	<Prorated item total>

Next: Purchase

Return transaction(s) **<Prorated return> 0.00**  
Purchase transaction  
Net cost of exchange **<Exchange Total>**

## 2. Select the VM product that you want to purchase and type a quantity. Make sure that the new purchase total is more than the return total. **Determine the right size before you purchase.**

Home > Reservations > Exchange

**Exchange**

Return Purchase Review

Select the reservation that you want to purchase. Purchase will be charged in full and will show

\* Name: VirtualMachines\_Reservation\_03-28-2019\_13...

Type: Virtual machine

\* Product: Standard\_DS1

vCPU (1), RAM (3.5)  
Region (South Central US)  
Change

Select the product you want to purchase

Reserved VM Instances (RIs) provide a significant discount over pay-as-you-go VM prices by allowing you to pre-purchase the base costs of your VM usage for a period of 1 or 3 years. Learn More

Scope: Shared Subscription: ExchangeTesting

Filter by name... Region: South Central US Term: One Year Add Filter Reset filters

NAME	REGION	TERM	SERIES	VCPUS	RAM	RECOMMENDED
Standard_B1s	South Central US	One Year	BS Series	1	1	1
Standard_DS1_v2	South Central US	One Year	DSv2 Series	1	3	1
Standard_B1ms	South Central US	One Year	BS Series	1	1	0
Standard_B1ms	South Central US	One Year	BS Series	1	2	0
Standard_B2s	South Central US	One Year	BS Series	2	4	0
Standard_B2ms	South Central US	One Year	BS Series	2	8	0
Standard_B4ms	South Central US	One Year	BS Series	4	16	0
Standard_B8ms	South Central US	One Year	BS Series	8	32	0
Standard_D1	South Central US	One Year	D Series	1	3	0
Standard_D2	South Central US	One Year	D Series	2	7	0
Standard_D11	South Central US	One Year	D Series	2	14	0
Standard_D3	South Central US	One Year	D Series	4	14	0
Standard_D12	South Central US	One Year	D Series	4	28	0
Standard_D4	South Central US	One Year	D Series	8	28	0
Standard_D13	South Central US	One Year	D Series	8	56	0
Standard_D14	South Central US	One Year	D Series	16	112	0
<b>Standard_DS1</b>	South Central US	One Year	DS Series	1	3	0
Standard_DS2	South Central US	One Year	DS Series	2	7	0
Standard_DS11	South Central US	One Year	DS Series	2	14	0
Standard_DS3	South Central US	One Year	DS Series	4	14	0

Previous Next: Review

Cost per unit: <Unit cost> <Estimated savings %>

### 3. Review and complete the transaction.

The screenshot shows the Azure portal's Exchange interface. The 'Review' tab is active. The 'RI to exchange' section lists two reservations: 'VirtualMachines\_Reservation\_03-26-2019\_07-07' and 'virtualMachines\_Reservation\_03-28-2019\_08-55'. The 'RI to purchase' section lists one reservation: 'VirtualMachines\_Reservation\_03-28-2019\_13-18'. Both sections include notes about separate transactions for returns and purchases.

To refund a reservation, go to **Reservation Details** and select **Refund**.

## Exchange multiple reservations

You can return similar types of reservations in one action.

When you exchange reservations, the new purchase currency amount must be greater than the refund amount. You can exchange any number of reservations for other allowed reservations if the currency amount is greater or equal to returned (exchanged) reservations. If your new purchase amount is less than the refund amount, an error message appears. If you see the error, reduce the quantity you want to return or increase the amount to purchase.

1. Sign in to the Azure portal and navigate to **Reservations**.
2. In the list of reservations, select the box for each reservation that you want to exchange.
3. At the top of the page, select **Exchange**.
4. If needed, revise the quantity to return for each reservation.
5. If you select the autofill return quantity, you can choose to **Refund all** to fill the list with the full quantity that you own for each reservation. Or, select **Optimize for utilization (7-day)** to fill the list with a quantity that optimizes for utilization based on the last seven days of usage. **Select Apply**.
6. At the bottom of the page, select **Next: Purchase**.
7. On the purchase tab, select the available products that you want to exchange for.  
You can select multiple products of different types.
8. In the Select the product you want to purchase pane, select the products you want and then select **Add to cart** and then select **Close**.
9. When done, select **Next: Review**.
10. Review your reservations to return and new reservations to purchase and then select **Confirm exchange**.

# Exchange nonpremium storage for premium storage

You can exchange a reservation purchased for a VM size that doesn't support premium storage to a corresponding VM size that does. For example, an *F1* for an *F1s*. To make the exchange, go to Reservation Details and select **Exchange**. The exchange doesn't reset the term of the reserved instance or create a new transaction. If you're exchanging for a different size, series, region or payment frequency, the term is reset for the new reservation.

## How transactions are processed

Microsoft cancels the existing reservation. Then the pro-rated amount for that reservation is refunded. If there's an exchange, the new purchase is processed. Microsoft processes refunds using one of the following methods, depending on your account type and payment method.

## Enterprise agreement customers

Money is added to the Azure Prepayment (previously called monetary commitment) for exchanges and refunds if the original purchase was made using one. If the Azure Prepayment term using the reservation was purchased is no longer active, then credit is added to your current enterprise agreement Azure Prepayment term. The credit is valid for 90 days from the date of refund. Unused credit expires at the end of 90 days.

If the original purchase was made as an overage, the original invoice on which the reservation was purchased and all later invoices are reopened and readjusted. Microsoft issues a credit memo for the refunds.

## Pay-as-you-go invoice payments and CSP program

The original reservation purchase invoice is canceled and then a new invoice is created for the refund. For exchanges, the new invoice shows the refund and the new purchase. The refund amount is adjusted against the purchase. If you only refunded a reservation, then the prorated amount stays with Microsoft and it's adjusted against a future reservation purchase. If you bought a reservation at pay-as-you-go rates and later move to a CSP, the reservation can be returned and repurchased without a penalty.

Although a CSP customer can't exchange, cancel, renew, or refund a reservation themselves, they can ask their partner to do it on their behalf.

## Pay-as-you-go credit card customers

The original invoice is canceled, and a new invoice is created. The money is refunded to the credit card that was used for the original purchase. If you've changed your card, [contact support](#).

## Cancel, exchange, and refund policies

Azure has the following policies for cancellations, exchanges, and refunds.

### Exchange policies

- You can return multiple existing reservations to purchase one new reservation of the same type. You can't exchange reservations of one type for another. For example, you can't return a VM reservation to purchase a SQL reservation. You can change a reservation property such as family, series, version, SKU, region, quantity, and term with an exchange.
- Only reservation owners can process an exchange. [Learn how to Add or change users who can manage a reservation](#).
- An exchange is processed as a refund and a repurchase – different transactions are created for the cancellation and the new reservation purchase. The prorated reservation amount is refunded for the reservations that's traded-in. You're charged fully for the new purchase. The prorated reservation amount is the daily prorated residual value of the reservation being returned.
- You can exchange or refund reservations even if the enterprise agreement used to purchase the reservation is expired and was renewed as a new agreement.
- The new reservation's lifetime commitment should equal or be greater than the returned reservation's remaining commitment. Example: for a three-year reservation that's \$100 per month and exchanged after the 18th payment, the new reservation's lifetime commitment should be \$1,800 or more (paid monthly or upfront).
- The new reservation purchased as part of exchange has a new term starting from the time of exchange.
- There's no penalty or annual limits for exchanges.

### Refund policies

- We're currently not charging an early termination fee, but in the future there might be a 12% early termination fee for cancellations.
- The total canceled commitment can't exceed 50,000 USD in a 12-month rolling window for a billing profile or single enrollment. For example, assume you have a three-year reservation (36 months). It costs 100 USD per month. It's refunded in

the 12th month. The canceled commitment is 2,400 USD (for the remaining 24 months). After the refund, your new available limit for refund is 47,600 USD (50,000-2,400). In 365 days from the refund, the 47,600 USD limit increases by 2,400 USD. Your new pool is 50,000 USD. Any other reservation cancellation for the billing profile or EA enrollment depletes the same pool, and the same replenishment logic applies.

- Azure doesn't process any refund that exceeds the 50,000 USD limit in a 12-month window for a billing profile or EA enrollment.
  - Refunds that result from an exchange don't count against the refund limit.
- Refunds are calculated based on the lowest price of either your purchase price or the current price of the reservation.
- Only reservation order owners can process a refund. [Learn how to Add or change users who can manage a reservation](#).
- For CSP program, the 50,000 USD limit is per customer.

Let's look at an example with the previous points in mind. If you bought a \$300,000 reservation, you can exchange it at any time for another reservation that equals or costs more (of the remaining reservation balance, not the original purchase price). For this example:

- There's no penalty or annual limits for exchanges.
- The refund that results from the exchange doesn't count against the refund limit.

## Need help? Contact us.

If you have questions or need help, [create a support request](#).

## Next steps

- To learn how to manage a reservation, see [Manage Azure Reservations](#).
- Learn about [Azure savings plan for compute](#)
- To learn more about Azure Reservations, see the following articles:
  - [What are Azure Reservations?](#)
  - [Manage Reservations in Azure](#)
  - [Understand how the reservation discount is applied](#)
  - [Understand reservation usage for your Pay-As-You-Go subscription](#)
  - [Understand reservation usage for your Enterprise enrollment](#)
  - [Windows software costs not included with reservations](#)
  - [Azure Reservations in the CSP program](#)

# Self-service exchanges and refunds for Azure Reservations

Article • 05/03/2023

Azure Reservations provide flexibility to help meet your evolving needs. Reservation products are interchangeable with each other if they're the same type of reservation. For example, you can exchange multiple compute reservations including Azure Dedicated Host, Azure VMware Solution, and Azure Virtual Machines with each other all at once. You can also exchange multiple SQL database reservation types including SQL Managed Instances and Elastic Pool with each other.

However, you can't exchange dissimilar reservations. For example, you can't exchange an Azure Cosmos DB reservation for SQL Database.

You can also exchange a reservation to purchase another reservation of a similar type in a different region. For example, you can exchange a reservation that's in West US 2 region for one that's in West Europe region.

## ⓘ Note

Exchanges will be unavailable for all compute reservations - Azure Reserved Virtual Machine Instances, Azure Dedicated Host reservations, and Azure App Services reservations - purchased on or after **January 1, 2024**. Compute reservations purchased **prior to January 1, 2024** will reserve the right to **exchange one more time** after the policy change goes into effect. Microsoft launched Azure savings plan for compute and it's designed to help you save broadly on predictable compute usage. The savings plan provides more flexibility needed to accommodate changes such as virtual machine series and regions. With savings plan providing the flexibility automatically, we're adjusting our reservations exchange policy. You can continue to use instance size flexibility for VM sizes, but we'll no longer support exchanging instance series or regions for Azure Reserved Virtual Machine Instances, Azure Dedicated Host reservations, and Azure App Services reservations. For more information about the exchange policy change, see [Changes to the Azure reservation exchange policy](#).

You may **trade-in** your Azure compute reservations for a savings plan. Or, you may continue to use and purchase reservations for those predictable, stable workloads where you know the specific configuration you'll need and want additional savings. Learn more about [Azure savings plan for compute and how it works with reservations](#).

When you exchange a reservation, you can change your term from one-year to three-year. Or, you can change the term from three-year to one-year.

You can also refund reservations, but the sum total of all canceled reservation commitment in your billing scope (such as EA, Microsoft Customer Agreement, and Microsoft Partner Agreement) can't exceed USD 50,000 in a 12 month rolling window.

The following reservations aren't eligible for refunds:

- Azure Databricks reserved capacity
- Synapse Analytics Pre-purchase plan
- Azure VMware solution by CloudSimple
- Azure Red Hat Open Shift
- Red Hat plans
- SUSE Linux plans

 **Note**

- You must have owner or Reservation administrator access on the Reservation Order to exchange or refund an existing reservation. You can Add or change users who can manage a reservation.
- Microsoft is not currently charging early termination fees for reservation refunds. We might charge the fees for refunds made in the future. We currently don't have a date for enabling the fee.

## How to exchange or refund an existing reservation

You can exchange your reservation from [Azure portal](#).

## 1. Select the reservations that you want to refund and select Exchange.

Home > Reservations > Exchange

**Exchange**

Return Purchase Review

Configure the reservations that you want to return. These reservations will be canceled immediately and the original charges will be reversed. It can take up to 7 business days for the charges to be refunded. Learn more about how refund and exchange are processed.

NAME	PRODUCT NAME	REGION	QUANTITY	PRORATED COST (PER UNIT)	QUANTITY TO RETURN	REFUND SUBTOTAL
VirtualMachines_Reservation_03-26-2019_07-07	Standard_B1s	South Central US	1	<Prorated cost>	1	<Prorated item total>
virtualMachines_Reservation_03-28-2019_08-55	Standard_B1s	South Central US	1	<Prorated cost>	1	<Prorated item total>

Next: Purchase

Return transaction(s) **<Prorated return> 0.00**  
Purchase transaction  
Net cost of exchange **<Exchange Total>**

## 2. Select the VM product that you want to purchase and type a quantity. Make sure that the new purchase total is more than the return total. **Determine the right size before you purchase.**

Home > Reservations > Exchange

**Exchange**

Return Purchase Review

Select the reservation that you want to purchase. Purchase will be charged in full and will show

\* Name: VirtualMachines\_Reservation\_03-28-2019\_13...

Type: Virtual machine

\* Product: Standard\_DS1

vCPU (1), RAM (3.5)  
Region (South Central US)  
Change

Select the product you want to purchase

Reserved VM Instances (RIs) provide a significant discount over pay-as-you-go VM prices by allowing you to pre-purchase the base costs of your VM usage for a period of 1 or 3 years. Learn More

Scope: Shared Subscription: ExchangeTesting

Filter by name... Region: South Central US Term: One Year Add Filter Reset filters

NAME	REGION	TERM	SERIES	VCPUS	RAM	RECOMMENDED
Standard_B1s	South Central US	One Year	BS Series	1	1	1
Standard_DS1_v2	South Central US	One Year	DSv2 Series	1	3	1
Standard_B1ms	South Central US	One Year	BS Series	1	1	0
Standard_B1ms	South Central US	One Year	BS Series	1	2	0
Standard_B2s	South Central US	One Year	BS Series	2	4	0
Standard_B2ms	South Central US	One Year	BS Series	2	8	0
Standard_B4ms	South Central US	One Year	BS Series	4	16	0
Standard_B8ms	South Central US	One Year	BS Series	8	32	0
Standard_D1	South Central US	One Year	D Series	1	3	0
Standard_D2	South Central US	One Year	D Series	2	7	0
Standard_D11	South Central US	One Year	D Series	2	14	0
Standard_D3	South Central US	One Year	D Series	4	14	0
Standard_D12	South Central US	One Year	D Series	4	28	0
Standard_D4	South Central US	One Year	D Series	8	28	0
Standard_D13	South Central US	One Year	D Series	8	56	0
Standard_D14	South Central US	One Year	D Series	16	112	0
<b>Standard_DS1</b>	South Central US	One Year	DS Series	1	3	0
Standard_DS2	South Central US	One Year	DS Series	2	7	0
Standard_DS11	South Central US	One Year	DS Series	2	14	0
Standard_DS3	South Central US	One Year	DS Series	4	14	0

Previous Next: Review

Cost per unit: **<Unit cost> <Estimated savings %>**

### 3. Review and complete the transaction.

The screenshot shows the Azure portal's Exchange interface. The top navigation bar includes Home, Reservations, Exchange, and a close button. Below the navigation is a breadcrumb trail: Home > Reservations > Exchange. The main content area has tabs: Return, Purchase, and Review, with Review being the active tab. The title "Exchange" is displayed above the content. The "RI to exchange" section lists two reservations: "VirtualMachines\_Reservation\_03-26-2019\_07-07" and "virtualMachines\_Reservation\_03-28-2019\_08-55", both categorized under "Standard\_B1s" in "South Central US". The "RI to purchase" section lists one reservation: "VirtualMachines\_Reservation\_03-28-2019\_13-18" under "Standard\_DS1" in "South Central US". Both sections include columns for Quantity, Prorated Cost (Per Unit), Quantity to Return, and Refund Subtotal. A note at the bottom of each section states: "Learn more about how refunds are processed" and "Note: Exchange leads to separate transactions for returns and purchases. You will be refunded the return amount and charged the purchase amount separately." At the bottom right of the page, there are "Return total: <Return Total>" and "Purchase total: <Purchase Total>" buttons.

To refund a reservation, go to **Reservation Details** and select **Refund**.

## Exchange multiple reservations

You can return similar types of reservations in one action.

When you exchange reservations, the new purchase currency amount must be greater than the refund amount. You can exchange any number of reservations for other allowed reservations if the currency amount is greater or equal to returned (exchanged) reservations. If your new purchase amount is less than the refund amount, an error message appears. If you see the error, reduce the quantity you want to return or increase the amount to purchase.

1. Sign in to the Azure portal and navigate to **Reservations**.
2. In the list of reservations, select the box for each reservation that you want to exchange.
3. At the top of the page, select **Exchange**.
4. If needed, revise the quantity to return for each reservation.
5. If you select the autofill return quantity, you can choose to **Refund all** to fill the list with the full quantity that you own for each reservation. Or, select **Optimize for utilization (7-day)** to fill the list with a quantity that optimizes for utilization based on the last seven days of usage. **Select Apply**.
6. At the bottom of the page, select **Next: Purchase**.
7. On the purchase tab, select the available products that you want to exchange for.  
You can select multiple products of different types.
8. In the Select the product you want to purchase pane, select the products you want and then select **Add to cart** and then select **Close**.
9. When done, select **Next: Review**.
10. Review your reservations to return and new reservations to purchase and then select **Confirm exchange**.

# Exchange nonpremium storage for premium storage

You can exchange a reservation purchased for a VM size that doesn't support premium storage to a corresponding VM size that does. For example, an *F1* for an *F1s*. To make the exchange, go to Reservation Details and select **Exchange**. The exchange doesn't reset the term of the reserved instance or create a new transaction. If you're exchanging for a different size, series, region or payment frequency, the term is reset for the new reservation.

## How transactions are processed

Microsoft cancels the existing reservation. Then the pro-rated amount for that reservation is refunded. If there's an exchange, the new purchase is processed. Microsoft processes refunds using one of the following methods, depending on your account type and payment method.

## Enterprise agreement customers

Money is added to the Azure Prepayment (previously called monetary commitment) for exchanges and refunds if the original purchase was made using one. If the Azure Prepayment term using the reservation was purchased is no longer active, then credit is added to your current enterprise agreement Azure Prepayment term. The credit is valid for 90 days from the date of refund. Unused credit expires at the end of 90 days.

If the original purchase was made as an overage, the original invoice on which the reservation was purchased and all later invoices are reopened and readjusted. Microsoft issues a credit memo for the refunds.

## Pay-as-you-go invoice payments and CSP program

The original reservation purchase invoice is canceled and then a new invoice is created for the refund. For exchanges, the new invoice shows the refund and the new purchase. The refund amount is adjusted against the purchase. If you only refunded a reservation, then the prorated amount stays with Microsoft and it's adjusted against a future reservation purchase. If you bought a reservation at pay-as-you-go rates and later move to a CSP, the reservation can be returned and repurchased without a penalty.

Although a CSP customer can't exchange, cancel, renew, or refund a reservation themselves, they can ask their partner to do it on their behalf.

## Pay-as-you-go credit card customers

The original invoice is canceled, and a new invoice is created. The money is refunded to the credit card that was used for the original purchase. If you've changed your card, [contact support](#).

## Cancel, exchange, and refund policies

Azure has the following policies for cancellations, exchanges, and refunds.

### Exchange policies

- You can return multiple existing reservations to purchase one new reservation of the same type. You can't exchange reservations of one type for another. For example, you can't return a VM reservation to purchase a SQL reservation. You can change a reservation property such as family, series, version, SKU, region, quantity, and term with an exchange.
- Only reservation owners can process an exchange. [Learn how to Add or change users who can manage a reservation](#).
- An exchange is processed as a refund and a repurchase – different transactions are created for the cancellation and the new reservation purchase. The prorated reservation amount is refunded for the reservations that's traded-in. You're charged fully for the new purchase. The prorated reservation amount is the daily prorated residual value of the reservation being returned.
- You can exchange or refund reservations even if the enterprise agreement used to purchase the reservation is expired and was renewed as a new agreement.
- The new reservation's lifetime commitment should equal or be greater than the returned reservation's remaining commitment. Example: for a three-year reservation that's \$100 per month and exchanged after the 18th payment, the new reservation's lifetime commitment should be \$1,800 or more (paid monthly or upfront).
- The new reservation purchased as part of exchange has a new term starting from the time of exchange.
- There's no penalty or annual limits for exchanges.

### Refund policies

- We're currently not charging an early termination fee, but in the future there might be a 12% early termination fee for cancellations.
- The total canceled commitment can't exceed 50,000 USD in a 12-month rolling window for a billing profile or single enrollment. For example, assume you have a three-year reservation (36 months). It costs 100 USD per month. It's refunded in

the 12th month. The canceled commitment is 2,400 USD (for the remaining 24 months). After the refund, your new available limit for refund is 47,600 USD (50,000-2,400). In 365 days from the refund, the 47,600 USD limit increases by 2,400 USD. Your new pool is 50,000 USD. Any other reservation cancellation for the billing profile or EA enrollment depletes the same pool, and the same replenishment logic applies.

- Azure doesn't process any refund that exceeds the 50,000 USD limit in a 12-month window for a billing profile or EA enrollment.
  - Refunds that result from an exchange don't count against the refund limit.
- Refunds are calculated based on the lowest price of either your purchase price or the current price of the reservation.
- Only reservation order owners can process a refund. [Learn how to Add or change users who can manage a reservation](#).
- For CSP program, the 50,000 USD limit is per customer.

Let's look at an example with the previous points in mind. If you bought a \$300,000 reservation, you can exchange it at any time for another reservation that equals or costs more (of the remaining reservation balance, not the original purchase price). For this example:

- There's no penalty or annual limits for exchanges.
- The refund that results from the exchange doesn't count against the refund limit.

## Need help? Contact us.

If you have questions or need help, [create a support request](#).

## Next steps

- To learn how to manage a reservation, see [Manage Azure Reservations](#).
- Learn about [Azure savings plan for compute](#)
- To learn more about Azure Reservations, see the following articles:
  - [What are Azure Reservations?](#)
  - [Manage Reservations in Azure](#)
  - [Understand how the reservation discount is applied](#)
  - [Understand reservation usage for your Pay-As-You-Go subscription](#)
  - [Understand reservation usage for your Enterprise enrollment](#)
  - [Windows software costs not included with reservations](#)
  - [Azure Reservations in the CSP program](#)

# Automatically renew reservations

Article • 12/07/2022

You can renew reservations to automatically purchase a replacement when an existing reservation expires. Automatic renewal provides an easy way to continue getting reservation discounts. It also saves you from having to closely monitor a reservation's expiration. With automatic renewal, you prevent savings benefits loss by not having to manually renew. The renewal setting is turned off by default. Enable or disable the renewal setting anytime, up to the expiration of the existing reservation.

Renewing a reservation creates a new reservation when the existing reservation expires. It doesn't extend the term of the existing reservation.

Opt in to automatically renew at any time. The renewal price is available 30 days before the expiry of existing reservation. When you enable renewal more than 30 days before the reservation expiration, you're sent an email detailing renewal costs 30 days before expiration. The reservation price might change between the time that you lock the renewal price and the renewal time. If so, your renewal will not be processed and you can purchase a new reservation in order to continue getting the benefit.

There's no obligation to renew and you can opt out of the renewal at any time before the existing reservation expires.

## Set up renewal

Go to Azure portal > **Reservations**.

1. Select the reservation.
2. Select **Renewal**.

### 3. Select Automatically purchase a new reservation upon expiry.

The screenshot shows the Azure portal interface for managing a reservation. The left sidebar has a tree view with 'Reservations' selected. The main area is titled '**<ReservationID> (<ReservationOrderID>) - Renewal**'. At the top right are 'Save' and 'Discard' buttons. Below them is a section titled 'Renewal' with the following details:

- Average utilization in the last 90 days: 100%
- Current reservation expiry date: 8/3/2019 (in 30 days)
- A new reservation will be purchased automatically on the current expiry date if you renew. If you choose not to renew, your resources won't be interrupted but you will be charged the pay-as-you-go rate. You can change or cancel your renewal anytime before expiry.

A checkbox labeled 'Automatically purchase a new reservation upon expiry' is checked. Below this is a section titled 'New reservation settings' with the following properties:

SKU	Standard_D1 - West US
Scope	Shared
Optimization	Instance size flexibility
Term	One Year
* Quantity	1

Below these settings are some estimated values:

Unit price	<UnitPrice>
Estimated renewal cost	<EstimatedRenewalCost>
Estimated savings	<EstimatedSavings> <Percent>

\*Additional taxes and fees may apply at the time of purchase.

## If you don't renew

Your services continue to run normally. You're charged pay-as-you-go rates for your usage after the reservation expires. If the reservation wasn't set for automatic renewal before expiration, you can't renew an expired reservation. To continue to receive savings, you can buy a new reservation.

## Required renewal permissions

The following conditions are required to renew a reservation:

- You must be an owner of the existing reservation.
- You must be an owner of the subscription if the reservation is scoped to a single subscription or resource group.
- You must be an owner of the subscription if it has a shared scope or management group scope.

## Default renewal settings

By default, the renewal inherits all properties except automatic renewal setting from the expiring reservation. A reservation renewal purchase has the same SKU, region, scope,

billing subscription, term, and quantity.

However, you can update the renewal reservation purchase quantity, billing frequency, and commitment term to optimize your savings.

## When the new reservation is purchased

A new reservation is purchased when the existing reservation expires. We try to prevent any delay between the two reservations. Continuity ensures that your costs are predictable, and you continue to get discounts.

## Changing parent reservation after setting renewal

If you make any of the following changes to the expiring reservation, the reservation renewal is canceled:

- Split
- Merge
- Transferring the reservation from one account to another
- Transferring the reservation from a WebDirect subscription to an enterprise agreement (EA) subscription, or any other purchase method
- Renew the enrollment

The new reservation inherits the scope and instance size flexibility setting from the expiring reservation during renewal.

## New reservation permissions

Azure copies the permissions from the expiring reservation to the new reservation. Additionally, the subscription account administrator of the reservation purchase has access to the new reservation.

## Potential renewal problems

Azure may not process the renewal if:

- Payment can't be collected
- A system error occurs during renewal
- The expiring SKU isn't active during renewal

- The EA is renewed into a different EA

You'll receive an email notification if any of the preceding conditions occur and the renewal is deactivated.

## Renewal notification

Renewal notification emails are sent 30 days before expiration and again on the expiration date. The sending email address is `azure-noreply@microsoft.com`. You might want to add the email address to your safe senders or allowlist.

Emails are sent to different people depending on your purchase method:

- Customers with EA subscriptions
  - Notifications are sent to the EA notification contacts, EA admin, reservation owners, and the reservation administrator.
- Customers with Microsoft Customer Agreement (Azure Plan)
  - Notifications are sent to the reservation owners and the reservation administrator.
- Cloud Solution Provider and new commerce partners
  - Emails are sent to the partner notification contact.
- Individual subscription customers with pay-as-you-go rates
  - Emails are sent to users who are set up as account administrators, reservation owners, and the reservation administrator.

## Next steps

- To learn more about Azure Reservations, see [What are Azure Reservations?](#)

# Group and allocate costs using tag inheritance

Article • 04/17/2023

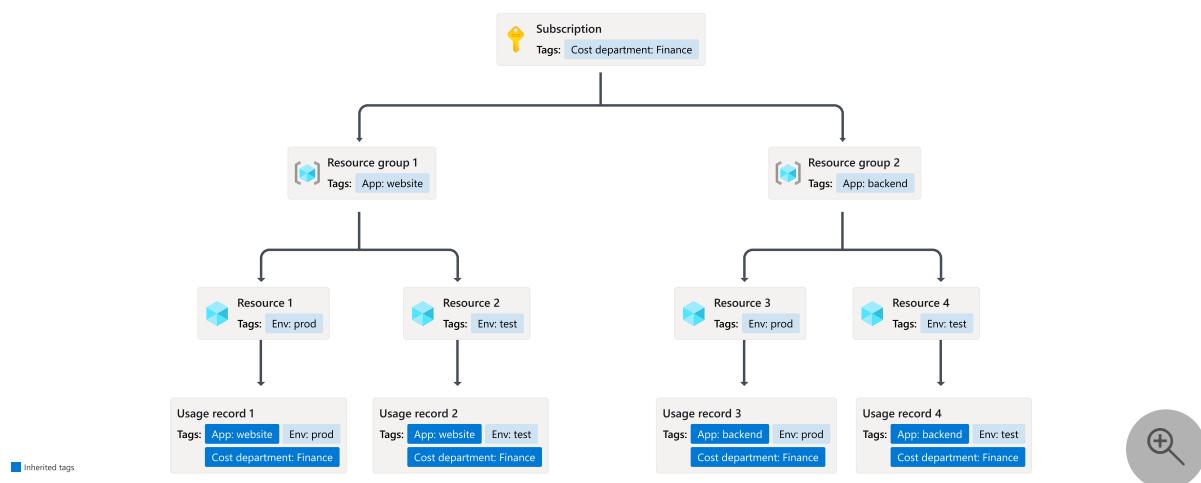
Azure tags are widely used to group costs to align with different business units, engineering environments, and cost departments. Tags provide the visibility needed for businesses to manage and allocate costs across the different groups.

This article explains how to use the tag inheritance setting in Cost Management. When enabled, tag inheritance applies resource group and subscription tags to child resource usage records. You don't have to tag every resource or rely on resources that emit usage to have their own tags.

Tag inheritance is available for the following billing account types:

- Enterprise Agreement (EA)
- Microsoft Customer Agreement (MCA)
- Microsoft Partner Agreement (MPA) with Azure plan subscriptions

Here's an example diagram showing how a tag is inherited. *Note that inherited tags are applied to child resource usage records and not the resources themselves.*



## Required permissions

- For subscriptions:
  - Cost Management reader to view
  - Cost Management Contributor to edit

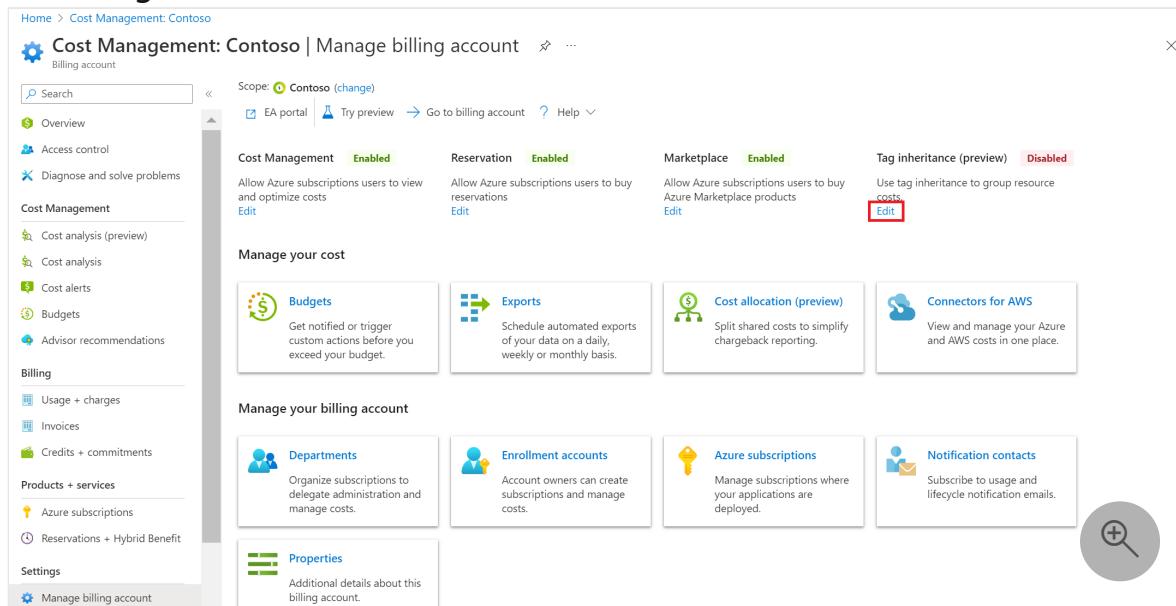
- For EA billing accounts:
  - Enterprise Administrator (read-only) to view
  - Enterprise Administrator to edit
- For MCA billing profiles:
  - Billing profile reader to view
  - Billing profile contributor to edit

## Enable tag inheritance

You can enable the tag inheritance setting in the Azure portal. You apply the setting at the EA billing account, MCA billing profile, and subscription scopes. After the setting is enabled, all resource group and subscription tags are automatically applied to child resource usage records.

### To enable tag inheritance in the Azure portal for an EA billing account

1. In the Azure portal, search for **Cost Management** and select it (the green hexagon-shaped symbol, *not* Cost Management + Billing).
2. Select a scope.
3. In the left menu under **Settings**, select **Manage billing account**.
4. Under **Tag inheritance**, select **Edit**.



5. In the Tag inheritance (Preview) window, select **Automatically apply subscription and resource group tags to new data**.

## To enable tag inheritance in the Azure portal for an MCA billing profile

1. In the Azure portal, search for **Cost Management** and select it (the green hexagon-shaped symbol, *not* Cost Management + Billing).
2. Select a scope.
3. In the left menu under **Settings**, select **Manage billing profile**.
4. Under **Tag inheritance**, select **Edit**.

5. In the Tag inheritance (Preview) window, select **Automatically apply subscription and resource group tags to new data**.

**Cost Management: Docs-test-billing-profile | Manage billing profile**

**Tag inheritance (preview)**

Contoso-billing-profile

Scope: Contoso Billing Account / Contoso-billing-profile (change)

Rename Try preview Go to billing profile Help

**Cost Management Enabled Reservation Enabled Marketplace**

Allow Azure subscriptions users to view and optimize costs Allow Azure subscriptions users to buy reservations Allow Azure Marketplace users to search and purchase products

**Manage your cost**

**Budgets** Get notified or trigger custom actions before you exceed your budget.

**Exports** Schedule automated exports of your data on a daily, weekly or monthly...

**Manage your billing profile**

**Invoice sections** Group subscriptions within your invoice and delegate subscription management.

**Azure subscriptions** Manage subscriptions where your applications are deployed.

**Payment methods** Manage payment methods used to pay your invoices.

**Properties** Additional details about this billing profile.

You may also be interested in

Inherited tags will start being applied within 24 hours and will apply to all data this month.

Apply Cancel Give feedback

## To enable tag inheritance in the Azure portal for a subscription

1. In the Azure portal, search for **Cost Management** and select it (the green hexagon-shaped symbol, *not* Cost Management + Billing).
2. Select a subscription scope.
3. In the left menu under **Settings**, select **Manage subscription**.
4. Under **Tag inheritance**, select **Edit**.

**Cost Management: Contoso subscription | Manage subscription**

**Tag inheritance (preview) Disabled**

Use tag inheritance to group resource costs.

**Manage your cost**

**Budgets** Get notified or trigger custom actions before you exceed your budget.

**Exports** Schedule automated exports of your data on a daily, weekly or monthly...

**Connectors for AWS** View and manage your Azure and AWS costs in one place.

**Manage your subscription**

**Management groups** Organize subscriptions into management groups to manage cost across...

**Resource groups** Manage applications and related resources deployed to resource groups.

**Resources** Manage individual resources to apply tags for reporting or resize to...

**Partner information** Partners can help deploy, manage, and optimize your cloud services.

5. In the Tag inheritance (Preview) window, select **Automatically apply subscription and resource group tags to new data**.

**Tag inheritance (preview)**

Contoso subscription

Use tag inheritance to group resource costs. [Learn more](#)

Automatically apply subscription and resource group tags to new usage data.

When the resource has a tag with the same name:

- Keep the resource tag
- Use the subscription or resource group tag

Inherited tags will start being applied within 24 hours and will apply to all data this month.

Apply Cancel Give feedback

## Choose between resource and inherited tags

When a resource tag matches the resource group or subscription tag being applied, the resource tag is applied to its usage record by default. You can change the default behavior to have the subscription or resource group tag override the resource tag.

In the Tag inheritance window, select the **Use the subscription or resource group tag** option.

**Tag inheritance (preview)**

Contoso

Use tag inheritance to group resource costs. [Learn more](#)

Automatically apply subscription and resource group tags to new usage data.

When the resource has a tag with the same name:

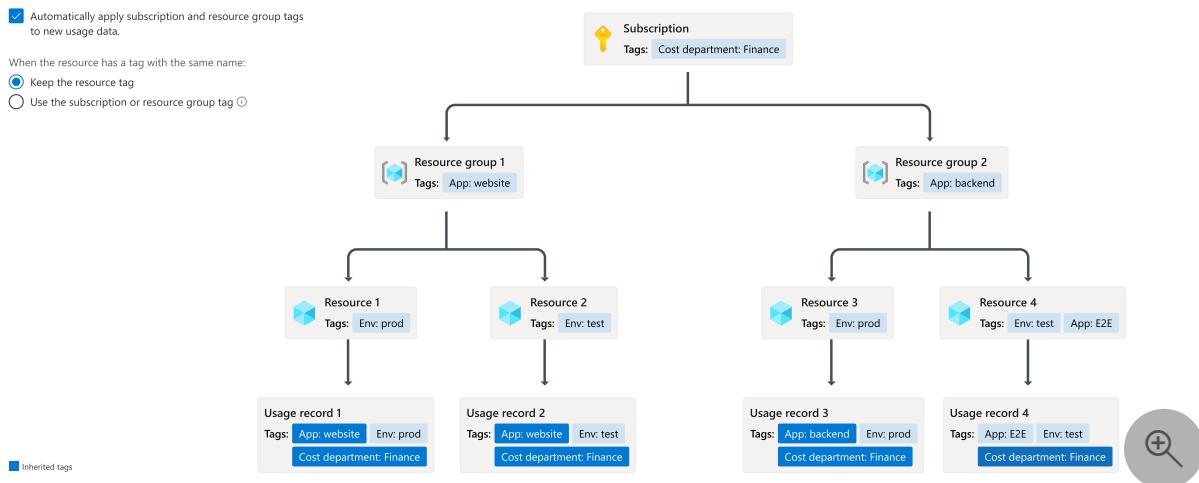
- Keep the resource tag
- Use the subscription or resource group tag

Inherited tags will start being applied within 24 hours and will apply to all data this month.

Apply Cancel Give feedback

Let's look at an example of how a resource tag gets applied. In the following diagram, resource 4 and resource group 2 have the same tag: *App*. Because the user chose to keep the resource tag, usage record 4 is updated with the resource tag value *E2E*.

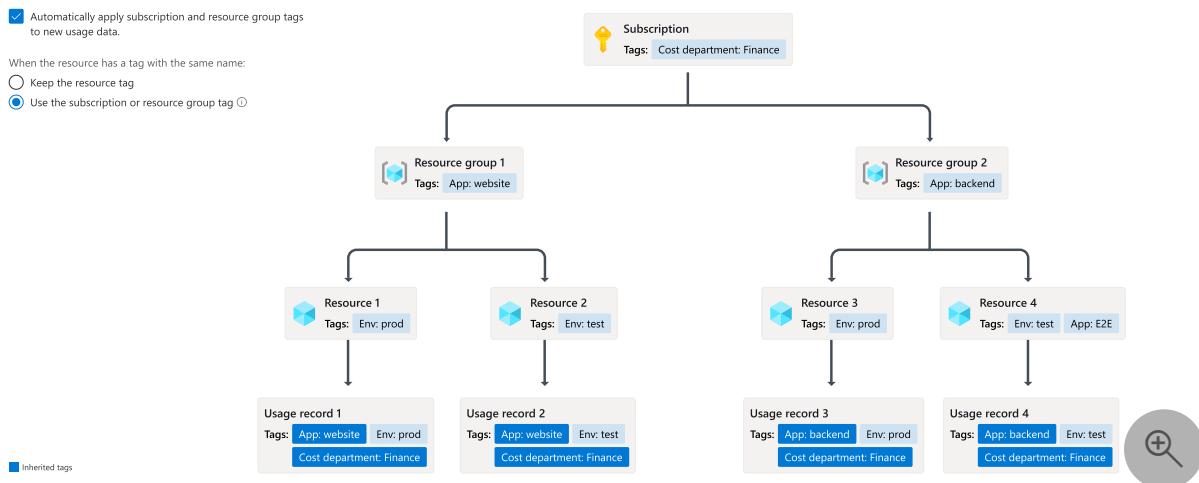
## Tag inheritance settings



Let's look at another example where a resource tag gets overridden. In the following diagram, resource 4 and resource group 2 have the same tag: **App**. Because the user chose to use the resource group or subscription tag, usage record 4 is updated with the resource group tag value, which is *backend*<sup>1</sup>.

<sup>1</sup> When the subscription and resource group tags are the same as the resource tag and you've selected the **Use the subscription or resource group tag** option, the subscription tag is used.

## Tag inheritance settings



# Usage record updates

After the tag inheritance setting is enabled, it takes about 8-24 hours for the child resource usage records to get updated with subscription and resource group tags. The usage records are updated for the current month using the existing subscription and resource group tags.

For example, if the tag inheritance setting is enabled on October 20, child resource usage records are updated from October 1 using the tags that existed on October 20.

Similarly, if the tag inheritance setting is disabled, the inherited tags will be removed from the usage records for the current month.

### ⓘ Note

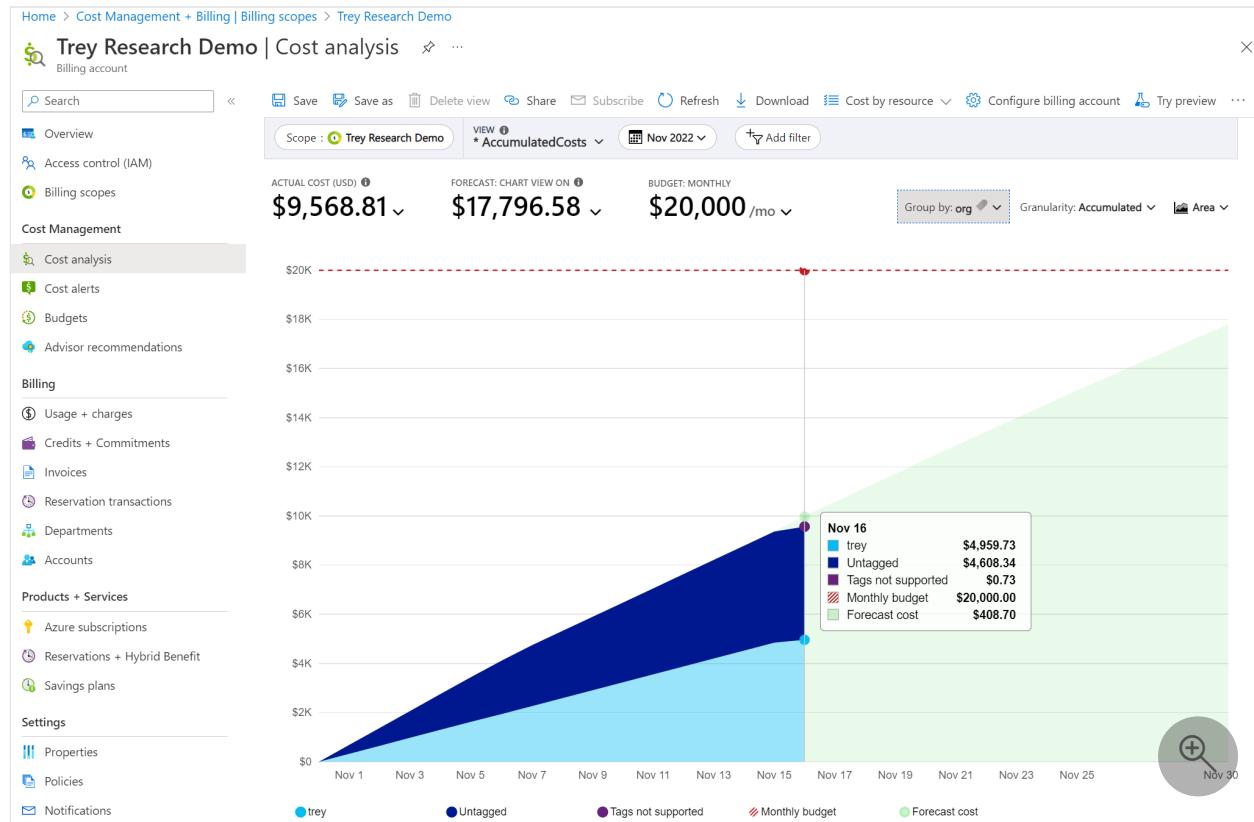
If there are purchases or resources that don't emit usage at a subscription scope, they will not have the subscription tags applied even if the setting is enabled.

## View costs grouped by tags

You can use cost analysis to view the costs grouped by tags.

1. In the Azure portal, navigate to **Cost Management**.
2. In the left menu, select **Cost Analysis**.
3. Select a scope.
4. In the **Group by** list, select the tag you want to view costs for.

Here's an example showing costs for the *org* tag.



You can also view the inherited tags by downloading your Azure usage. For more information, see [View and download your Azure usage and charges](#).

## Next steps

- Learn how to [split shared costs](#).

# Create and manage Azure cost allocation rules (Preview)

Article • 03/28/2023

Large enterprises often centrally manage Azure services or resources. However, different internal departments or business units use them. Typically, the centrally managing team wants to reallocate the cost of the shared services back out to the internal departments or organizational business units who are actively using the services. This article helps you understand and use cost allocation in Cost Management.

With cost allocation, you can reassign or distribute the costs of shared services. Costs from subscriptions, resource groups, or tags get assigned to other subscriptions, resource groups or tags in your organization. Cost allocation shifts costs of the shared services to another subscription, resource groups, or tags owned by the consuming internal departments or business units. In other words, cost allocation helps to manage and show *cost accountability* from one place to another.

Cost allocation doesn't affect your billing invoice. Billing responsibilities don't change. The primary purpose of cost allocation is to help you charge back costs to others. All chargeback processes happen in your organization outside of Azure. Cost allocation helps you charge back costs by showing them as the get reassigned or distributed.

Allocated costs appear in cost analysis. They appear as other items associated with the targeted subscriptions, resource groups, or tags that you specify when you create a cost allocation rule.

## ⓘ Note

Cost Management's cost allocation feature is currently in public preview. Some features in Cost Management might not be supported or might have limited capabilities.

## Prerequisites

- Cost allocation currently only supports customers with:
  - A [Microsoft Customer Agreement](#) (MCA) in the Enterprise motion where you buy Azure services through a Microsoft representative. It's also called an MCA enterprise agreement.

- A [Microsoft Customer Agreement](#) that you bought through the Azure website. It's also called an MCA individual agreement.
- An [Enterprise Agreement \(EA\)](#).
- To create or manage a cost allocation rule, you must use an Enterprise Administrator account for [Enterprise Agreements](#). Or you must be a [Billing account](#) owner for Microsoft Customer Agreements.

## Create a cost allocation rule

1. Sign in to the Azure portal at <https://portal.azure.com>.
2. Navigate to **Cost Management + Billing > Cost Management**.
3. Under **Settings > Configuration**, select **Cost allocation (Preview)**.
4. Ensure that you select the correct EA enrollment or billing account.
5. Select **+Add**.
6. Enter descriptive text for the cost allocation rule name.

The screenshot shows a form for creating a new cost allocation rule. The 'Name' field is filled with 'MyFirstAllocationRule'. Below the field, there is an informational message: 'The cost used in this rule's evaluation started on Mon Jun 01 2020. [Learn more](#)'.

The rule's evaluation start date generates the cost allocation percentages and prefills them.

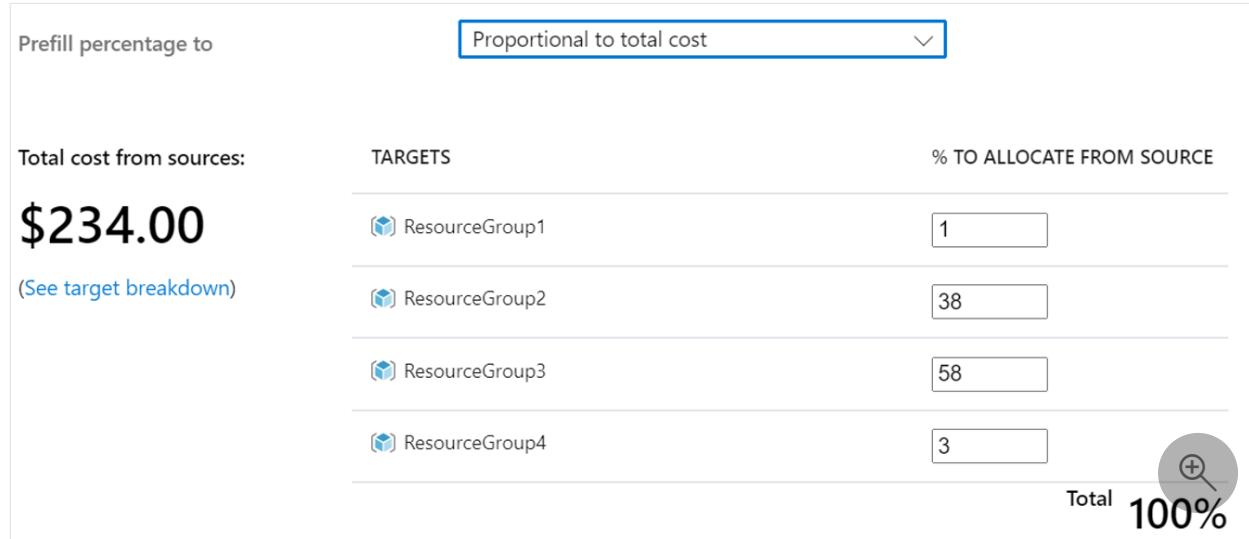
1. Select **Add sources** and then select either subscriptions, resource groups, or tags to choose costs to distribute.
2. Select **Add targets** and then select either subscriptions, resource groups, or tags to receive the allocated costs.
3. If you need to create more cost allocation rules, repeat this process.

## Configure the allocation percentage

Configure the allocation percentage to define how costs proportionally divide between the specified targets. You can manually define whole number percentages to create an allocation rule. Or you can split the costs proportionally based on the current usage of the compute, storage, or network across the specified targets.

When you distribute costs by compute cost, storage cost, or network cost, the proportional percentage is derived by evaluating the selected target's costs. The costs are associated with the resource type for the current billing month.

When you distribute costs proportional to total cost, the proportional percentage allocates by the sum or total cost of the selected targets for the current billing month.



Once set, the prefilled percentages defined don't change. All ongoing allocations use them. The percentages change only when you manually update the rule.

1. Select one of the following options in the **Prefill percentage to** list.

- **Distribute evenly** – Each of the targets receives an even percentage proportion of the total cost.
- **Total cost** – Creates a ratio proportional to the targets based on their total cost. It uses the ratio to distribute costs from the selected sources.
- **Compute cost** - Creates a ratio proportional to the targets based on their Azure compute cost (resource types in the [Microsoft.Compute](#) namespace). It uses the ratio to distribute costs from the selected sources.
- **Storage cost** - Creates a ratio proportional to the targets based on their Azure storage cost (resource types in the [Microsoft.Storage](#) namespace). It uses the ratio to distribute costs from the selected sources.
- **Network cost** - Creates a ratio proportional to the targets based on their Azure network cost (resource types in the [Microsoft.Network](#) namespace). It uses the ratio to distribute costs from the selected sources.
- **Custom** – Allows you to manually specify a whole number percentage. The specified total must equal 100%.

2. When done, select **Create**.

The allocation rule starts processing. When the rule is active, all the selected source's costs allocate to the specified targets.

### Note

New rule processing can take up to two hours before it completes and is active.

Here's a video that demonstrates how to create a cost allocation rule.

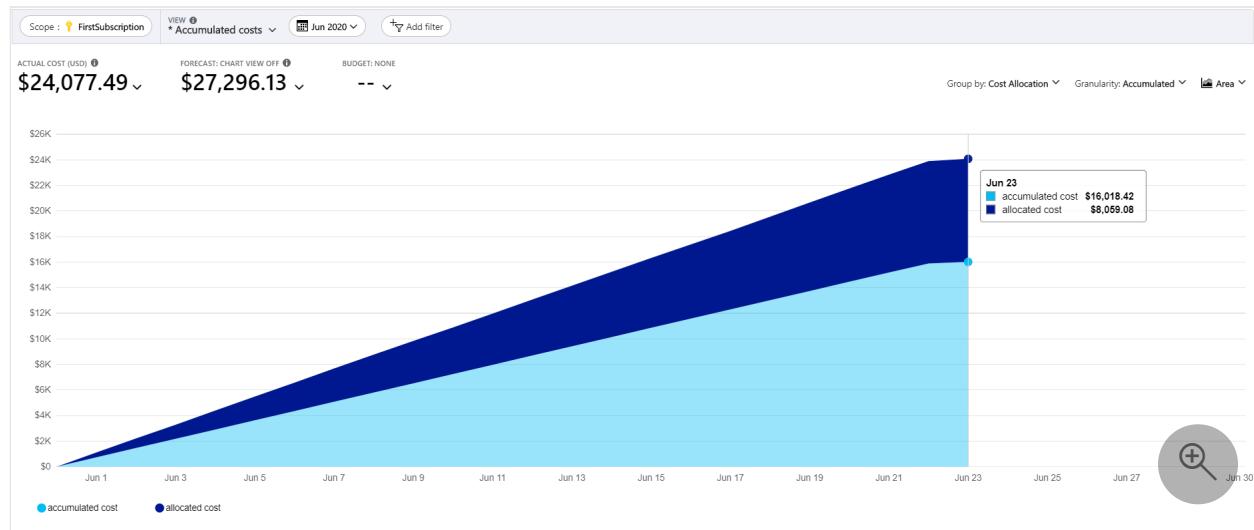
<https://www.youtube-nocookie.com/embed/nYzIls2mx9Q>

## Verify the cost allocation rule

When the cost allocation rule is active, costs from the selected sources distribute to the specified allocation targets. Use the following information to verify proper cost allocation to targets.

### View cost allocation for a subscription

You view the effect of the allocation rule in cost analysis. In the Azure portal, go to [Subscriptions](#). Select a subscription in the list that is the target of an active cost allocation rule. Then select **Cost analysis** in the menu. In Cost analysis, select **Group by** and then select **Cost allocation**. The resulting view shows a quick cost breakdown generated by the subscription. Costs allocated to the subscription appear, similar to the following image.

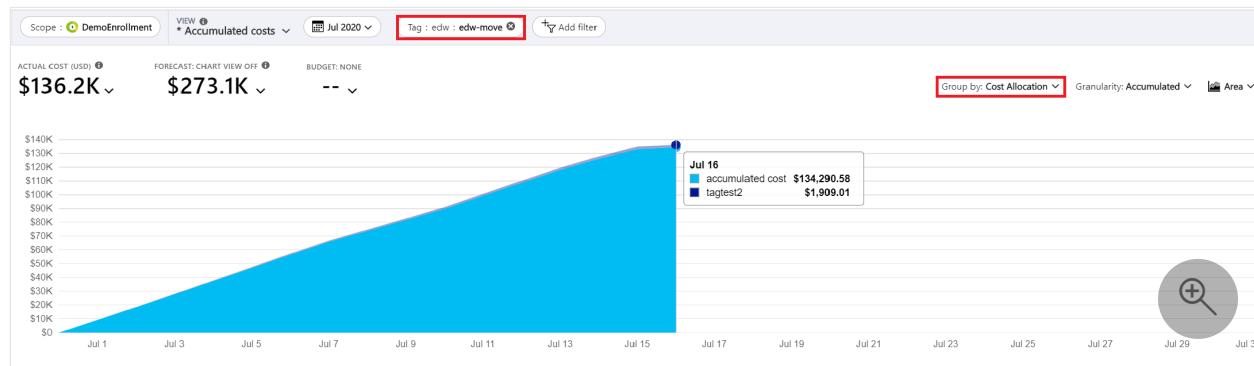


### View cost allocation for a resource group

Use a similar process to assess the effect of a cost allocation rule for a resource group. In the Azure portal, go to [Resource groups](#). Select a resource group in the list that an active cost allocation rule targets. Then select **Cost analysis** in the menu. In Cost analysis, select **Group by** and then select **Cost allocation**. The view shows you a quick cost breakdown generated by the resource group. It also shows cost allocated to the resource group.

## View cost allocation for tags

In the Azure portal, navigate to **Cost Management + Billing > Cost Management > Cost analysis**. In Cost analysis, select **Add filter**. Select **Tag**, choose the tag key, and tag values that have cost allocated to them.



## View cost allocation in the downloaded Usage Details and in Exports CSV files

Cost allocation rules are also available in the downloaded Usage Details file and in the exported data. The data files have the column name `costAllocationRuleName`. If a Cost allocation rule is applicable to an entry in Usage Details or Exports file, it populates the row with the Cost allocation rule name. The following example image shows a negative charge with an entry for the source subscription. It's the charge getting allocated cost from. There's also a positive charge for the Cost allocation rule's target.

date	product	subscriptionid	subscriptionName	paygCostInUsd	costAllocationRuleName
5/5/2021	Virtual Machines BS Series Windows - B1s - US East	11111111-1111-1111-1111-111111111111	ContosoSub1	-0.336	MyFirstAllocationRule
5/5/2021	Virtual Machines BS Series Windows - B1s - US East	22222222-2222-2222-2222-222222222222	ContosoSub2	0.336	MyFirstAllocationRule

## Azure invoice reconciliation

Azure invoice reconciliation also uses the Usage Details file. Showing any internal allocated costs during reconciliation could be confusing. To reduce any potential confusion and to align to the data shown on the invoice, you can filter out any Cost allocation rules. After you remove the cost allocation rules, your Usage Details file should match the cost shown by the billed subscription invoice.

date	product	subscriptionid	subscriptionName	paygCostInUsd	costAllocationRuleName
5/5/2021	Virtual Machines BS Series Windows - B1s - US East	11111111-1111-1111-1111-111111111111	ContosoSub1	0.336	

## Edit an existing cost allocation rule

You can edit a cost allocation rule to change the source or the target or if you want to update the prefilled percentage for either compute, storage, or network options. Edit the rules in the same way you create them. Modifying existing rules can take up to two hours to reprocess.

## Current limitations

Currently, Cost Management supports cost allocation in Cost analysis, budgets, and forecast views. Allocated costs appear in the subscriptions list and on the Subscriptions overview page.

The following items are currently unsupported by the cost allocation public preview:

- Billing subscriptions area
- [Cost Management Power BI App](#)
- [Power BI Desktop connector](#)

The [Usage Details API](#) version `2021-10-01` and later supports cost allocation data.

However, cost allocation data results might be empty if you're using an unsupported API or if you don't have any cost allocation rules.

If you have cost allocation rules enabled, the `UnitPrice` field in your usage details file is 0. We recommend that you use price sheet data to get unit price information until it's available in the usage details file.

Cost allocation to a target won't happen if that target doesn't have any costs associated with it.

## Next steps

- Read the [Cost Management + Billing FAQ](#) for questions and answers about cost allocation.
- Create or update allocation rules using the [Cost allocation REST API](#)
- Learn more about [How to optimize your cloud investment with Cost Management](#)

# Cost Allocation Rules

Reference

Service: Cost Management

API Version: 2020-03-01-preview

## Operations

<a href="#">Check Name Availability</a>	Checks availability and correctness of a name for a cost allocation rule
<a href="#">Create Or Update</a>	Create/Update a rule to allocate cost between different resources within a billing account or enterprise enrollment.
<a href="#">Delete</a>	Delete cost allocation rule for billing account or enterprise enrollment.
<a href="#">Get</a>	Get a cost allocation rule by rule name and billing account or enterprise enrollment.
<a href="#">List</a>	Get the list of all cost allocation rules for a billing account or enterprise enrollment.

# Settings

Reference

Service: Cost Management

API Version: 2022-10-01-preview

## Operations

<a href="#">Create Or Update By Scope</a>	Create or update a setting within the given scope.
<a href="#">Delete By Scope</a>	Delete a setting within the given scope.
<a href="#">Get By Scope</a>	Get the setting from the given scope by name.
<a href="#">List</a>	List all cost management settings in the requested scope.

# Cost Management automation overview

Article • 07/17/2022

You can use Cost Management automation and reporting to build a custom set of solutions to retrieve and manage cost data. This article covers what APIs are available for use and common scenarios for Cost Management automation.

## Available APIs

There are many different APIs that can be used to interact with Cost Management data. A summary of the available APIs and what they do is below. Multiple APIs may need to be used to achieve a specific scenario. Review the common scenarios outlined later to learn more.

For contractual information about how to call each API, review the API specification articles.

## Cost Details APIs

The APIs below provide you with cost details data (formerly referred to as usage details). Cost Details are the most granular usage and cost records that are available to you within the Azure ecosystem. All Cost Management experiences in the Azure portal and the APIs are built upon the raw dataset. To learn more, see [cost details overview](#).

- [Exports API](#) - Configure a recurring task to export your cost details data to Azure storage on a daily, weekly or monthly basis. Exported data is in CSV format. It's our recommended solution for ingesting cost data and is the most scalable for large enterprises. To learn more, see [Retrieve large cost datasets with exports](#).
- [Generate Cost Details](#) - Download a cost details CSV file on demand. It's useful for smaller, date range based datasets. For larger workloads, we strongly recommend that you use Exports. To learn more about using this API, see [Get small cost datasets on demand](#).

## Pricing APIs

- [Azure Retail Prices](#) - Get meter rates with pay-as-you-go pricing. You can use the returned information with your resource usage information to manually calculate the expected bill.

- [Price Sheet API](#) - Get custom pricing for all meters. Enterprises can use this data in combination with usage details and marketplace usage information to manually calculate costs by using usage and marketplace data.

## Budgets and Alerts APIs

- [Budgets API](#) - Create either cost budgets for resources, resource groups, or billing meters. When you've created budgets, you can configure alerts to notify you when you've exceeded defined budget thresholds. You can also configure actions to occur when you've reached budget amounts. For more information, see [Automate budget creation](#) and [Configure budget based actions](#).
- [Alerts API](#) - Manage all of the alerts that have been created by budgets and other Azure alerting systems.

## Invoicing APIs

- [Invoices API](#) - Get list of invoices. The API returns a summary of your invoices including total amount, payment status and a link to download a pdf copy of your invoice.
- [Transactions API](#) - Get invoice line-items for an invoice. You can use the API to get all purchases, refunds and credits that are included in your invoice. The API is only available for customers with Microsoft Customer Agreement or Microsoft Partner Agreement billing accounts.

## Reservation APIs

- [Reservation Details API](#) - Get the detailed resource consumption associated with your reservation purchases.
- [Reservation Transactions API](#) - Get reservation related purchase and management transactions.
- [Reservation Recommendations API](#) - Get recommendations for reservation purchases to make in the future along with expected savings information.
- [Reservation Recommendation Details API](#) - Get detailed information for specific reservation purchases to perform a what-if analysis.

## Common API scenarios

You can use the billing and cost management APIs in many scenarios to answer cost-related and usage-related questions. Common scenarios and how to use the different APIs to achieve those scenarios are outlined below.

## Invoice reconciliation

This scenario is used to address the following questions:

- Did Microsoft charge me the right amount on my invoice?
- What's my bill, and can I calculate it myself using the raw data?

To answer these questions, follow the steps below.

1. Call the [Invoices API](#) to get the info needed to download an invoice. If you're a Microsoft Customer Agreement customer and just wish to get the specific line items seen on your invoice automatically, you can also utilize the [Transactions API](#) to get those line items in an API-readable format.
2. Use either [Exports](#) or the [Cost Details API](#) to download the raw usage file.
3. Analyze the data in the raw usage file to compare it against the costs that are present on the invoice. For Azure consumption, the data in your invoice is rolled up based on the meter associated with your usage.

## Cross-charging

Once there's a good understanding of spending for a given month, organizations next need to determine what teams or divisions need to pay for the various charges accrued. Follow the steps below.

1. Use either [Exports](#) or the [Cost Details API](#) to download the raw usage file.
2. Analyze the data in the raw usage file and allocate it based on the organizational hierarchy that you have in place. Allocation could be based on resource groups, subscriptions, cost allocation rules, tags or other Azure organization hierarchies.
  - To learn more about best practices to consider when configuring your Azure environments, see [Cost management best practices](#).
  - To learn more about the scopes and the organizational structures available to you, see [Understand and work with scopes](#).
  - To set up allocation directly in Azure, see [Allocate costs](#).

## Azure spending prior to invoice closure

It's important to keep tabs on how costs are accruing throughout the month. Proactive analysis before the invoice is closed can provide opportunities to change spending patterns and get an invoice's projected costs down. To ingest all of the raw data that has accrued month-to-date, use [Exports API](#).

Configuring automatic alerting can also ensure that spending doesn't unexpectedly get out of hand and removes the need for manual cost monitoring throughout the month. To ensure your costs don't breach thresholds or aren't forecasted to breach thresholds, use the [Budgets API](#).

## Cost trend reporting

Often it's useful to understand how much an organization is spending over time. Understanding cost over time helps identify trends and areas for cost optimization improvement. Follow the steps below to set up a cost dataset that can be used for reporting cost over time at scale.

1. Extract the historical costs for prior months. See [Seed a historical cost dataset with the Exports API](#) to learn more.
2. Ingest your historical data from the Azure storage account associated with your Exports into a queryable store. We recommend SQL or Azure Synapse.
3. Configure a month-to-date Export to storage at a scope with the costs that need to be analyzed. Export to storage is done in the Azure portal. See [Export costs](#). The month-to-date Export will be used to properly extract costs moving forward.
4. Configure a data pipeline to ingest cost data for the open month into your queryable store. This pipeline should be used with the month-to-date Export that you've configured. Azure Data Factory provides good solutions for this kind of ingestion scenario.
5. Perform reporting as needed using reports built with your queryable store. Power BI can be good for this scenario. If you're looking for a more out of the box solution, see our [Power BI Template App](#).

## Reservation related investigations

For more information about reservation-specific automation scenarios, see [APIs for Azure reservation automation](#).

## Next steps

- To learn more about how to assign the proper permissions to call our APIs programmatically, see [Assign permissions to Cost Management APIs](#).

- To learn more about working with cost details, see [Ingest usage details data](#).
- To learn more about budget automation, see [Automate budget creation](#).
- For information about using REST APIs retrieve prices for all Azure services, see [Azure Retail Prices overview](#).
- To compare your invoice with the detailed daily usage file and the cost management reports in the Azure portal, see [Understand your bill for Microsoft Azure](#).
- If you have questions or need help, [create a support request](#).

# Tutorial: Create and manage exported data

Article • 03/14/2023

If you read the Cost Analysis tutorial, then you're familiar with manually downloading your Cost Management data. However, you can create a recurring task that automatically exports your Cost Management data to Azure storage on a daily, weekly, or monthly basis. Exported data is in CSV format and it contains all the information that's collected by Cost Management. You can then use the exported data in Azure storage with external systems and combine it with your own custom data. And you can use your exported data in an external system like a dashboard or other financial system.

Watch the [How to schedule exports to storage with Cost Management](#) video about creating a scheduled export of your Azure cost data to Azure Storage. To watch other videos, visit the [Cost Management YouTube channel](#).

[https://www.youtube-nocookie.com/embed/rWa\\_xl1aRzo](https://www.youtube-nocookie.com/embed/rWa_xl1aRzo)

The examples in this tutorial walk you through exporting your cost management data and then verify that the data was successfully exported.

In this tutorial, you learn how to:

- ✓ Create a daily export
- ✓ Verify that data is collected

## Prerequisites

Data export is available for various Azure account types, including [Enterprise Agreement \(EA\)](#) and [Microsoft Customer Agreement](#) customers. To view the full list of supported account types, see [Understand Cost Management data](#). The following Azure permissions, or scopes, are supported per subscription for data export by user and group. For more information about scopes, see [Understand and work with scopes](#).

- Owner - Can create, modify, or delete scheduled exports for a subscription.
- Contributor - Can create, modify, or delete their own scheduled exports. Can modify the name of scheduled exports created by others.
- Reader - Can schedule exports that they have permission to.
  - [For more information about scopes, including access needed to configure exports for Enterprise Agreement and Microsoft Customer agreement scopes, see Understand and work with scopes](#).

For Azure Storage accounts:

- Write permissions are required to change the configured storage account, independent of permissions on the export.
- Your Azure storage account must be configured for blob or file storage.
- The storage account must not have a firewall configured.
- The storage account configuration must have the **Permitted scope for copy operations (preview)** option set to **From any storage account**.

Contoso-storage-account | Configuration

Storage account | Directory: Microsoft

Search Save Discard Refresh

Account kind: Storage (general purpose v1)

This account can be upgraded to a General Purpose v2 account with additional features. Upgrading is permanent and will result in billing changes. Upgrade is currently disabled due to lack of write permission ('Microsoft.Storage/storageAccounts/write') on the storage account. [Learn more about upgrading storage accounts](#)

Upgrade

Performance: Standard

Secure transfer required: Enabled

Allow Blob public access: Disabled

Allow storage account key access: Enabled

Allow recommended upper limit for shared access signature (SAS) expiry interval: Enabled

Default to Azure Active Directory authorization in the Azure portal: Enabled

Minimum TLS version: Version 1.0

Permitted scope for copy operations: From any storage account

From storage accounts in the same Azure AD tenant

Large file shares: Disabled

If you have a new subscription, you can't immediately use Cost Management features. It might take up to 48 hours before you can use all Cost Management features.

## Sign in to Azure

Sign in to the Azure portal at <https://portal.azure.com>.

## Create a daily export

Portal

To create or view a data export or to schedule an export, choose a scope in the Azure portal and select **Cost analysis** in the menu. For example, navigate to **Subscriptions**, select a subscription from the list, and then select **Cost analysis** in the menu. At the top of the Cost analysis page, select **Settings**, then **Exports**.

**① Note**

- Besides subscriptions, you can create exports on resource groups, management groups, departments, and enrollments. For more information about scopes, see [Understand and work with scopes](#).
- When you're signed in as a partner at the billing account scope or on a customer's tenant, you can export data to an Azure Storage account that's linked to your partner storage account. However, you must have an active subscription in your CSP tenant.

1. Select **Add** and type a name for the export.

2. For the **Metric**, make a selection:

- **Actual cost (Usage and Purchases)** - Select to export standard usage and purchases
- **Amortized cost (Usage and Purchases)** - Select to export amortized costs for purchases like Azure reservations and Azure savings plan for compute.

3. For **Export type**, make a selection:

- **Daily export of month-to-date costs** - Provides a new export file daily for your month-to-date costs. The latest data is aggregated from previous daily exports.
- **Weekly export of cost for the last seven days** - Creates a weekly export of your costs for the past seven days from the selected start date of your export.
- **Monthly export of last month's costs** - Provides you with an export of your last month's costs compared to the current month that you create the export. Afterward, the schedule runs an export on the fifth day of every new month with your previous months costs.
- **One-time export** - Allows you to choose a date range for historical data to export to Azure blob storage. You can export a maximum of 90 days of historical costs from the day you choose. This export runs immediately and is available in your storage account within two hours. Depending on

your export type, either choose a start date, or choose a **From** and **To** date.

4. Specify the subscription for your Azure storage account, then select a resource group or create a new one.
5. Select the storage account name or create a new one.
6. Select the location (Azure region).
7. Specify the storage container and the directory path that you'd like the export file to go to.

The screenshot shows the Microsoft Azure portal with the URL [https://portal.azure.com/?feature.arm\\_canary=true&feature.exportapivers...](https://portal.azure.com/?feature.arm_canary=true&feature.exportapivers...). The page is titled 'New export' under 'Cost Management: Contoso (Demo) | Cost analysis > Configuration > Exports'. The 'Export details' section includes fields for Name (DemoExport), Metric (Actual cost (Usage and Purchases)), Export type (Daily export of month-to-date costs), and Start date (Wed Aug 05 2020). The 'Storage' section shows 'Subscription' (Trey Research Corporate), 'Resource group' (TreyNetwork), 'Account name' (cmdemo.core.windows.net), 'Location' ((US) East US), 'Container' (democontainer), and 'Directory' (demodirectory). A 'Create' button is at the bottom left, and a search icon is at the bottom right.

8. Review your export details and select **Create**.

Your new export appears in the list of exports. By default, new exports are enabled. If you want to disable or delete a scheduled export, select any item in the list, and then select either **Disable** or **Delete**.

Initially, it can take 12-24 hours before the export runs. However, it can take up longer before data is shown in exported files.

## Export schedule

Scheduled exports are affected by the time and day of week of when you initially create the export. When you create a scheduled export, the export runs at the same frequency for each export that runs later. For example, for a daily export of month-to-date costs export set at a daily frequency, the export runs once each UTC day. Similarly for a weekly export, the export runs every week on the same UTC day as it is scheduled.

Individual export runs can occur at different times throughout the day. So, avoid taking a firm dependency on the exact time of the export runs. Run timing depends on the active load present in Azure during a given UTC day. When an export run begins, your data should be available within 4 hours.

Exports are scheduled using Coordinated Universal Time (UTC). The Exports API always uses and displays UTC.

- When you create an export using the [Exports API](#), specify the `recurrencePeriod` in UTC time. The API doesn't convert your local time to UTC.
  - Example - A weekly export is scheduled on Friday, August 19 with `recurrencePeriod` set to 2:00 PM. The API receives the input as 2:00 PM UTC, Friday, August 19. The weekly export will be scheduled to run every Friday.
- When you create an export in the Azure portal, its start date time is automatically converted to the equivalent UTC time.
  - Example - A weekly export is scheduled on Friday, August 19 with the local time of 2:00 AM IST (UTC+5:30) from the Azure portal. The API receives the input as 8:30 PM, Thursday, August 18. The weekly export will be scheduled to run every Thursday.

Each export creates a new file, so older exports aren't overwritten.

## Create an export for multiple subscriptions

If you have an Enterprise Agreement, then you can use a management group to aggregate subscription cost information in a single container. Then you can export cost management data for the management group. When you create an export in the Azure portal, select the **Actual Costs** option. When you create a management group export using the API, create a *usage export*. Currently, exports at the management group scope only support usage charges. Purchases including reservations and savings plans aren't present in your exports file.

Exports for management groups of other subscription types aren't supported.

1. If you haven't already created a management group, create one group and assign subscriptions to it.
2. In cost analysis, set the scope to your management group and select **Select this management group**.

The screenshot shows the Microsoft Azure portal with the URL [https://portal.azure.com/?feature.arm\\_canary=true&feature.exportapiversion=2020-05-01-preview&...](https://portal.azure.com/?feature.arm_canary=true&feature.exportapiversion=2020-05-01-preview&...). The user is logged in as admin@contoso.com (CONTOSO). The main page is 'Cost Management: Contoso (Demo) | Configuration'. On the left, there's a sidebar with 'Cost Management' selected, showing options like 'Cost analysis', 'Cost alerts', 'Budgets', etc. The main area shows 'View and manage' sections for 'Departments' and 'Subscription features'. A 'Select scope' dialog is open over the page, with the 'Scope' dropdown set to 'Contoso (Demo)' and the 'Select this management group' button highlighted with a red box.

3. Create an export at the scope to get cost management data for the subscriptions in the management group.

The screenshot shows the Microsoft Azure portal with the URL <https://portal.azure.com/#blade/HubsBlade/resourceType=Microsoft.CostManagement%2FcostManagementGroups%2F%7BresourceId%7D/resourceType=Microsoft.CostManagement%2Fexports%2F%7BresourceId%7D>. The user is logged in as admin@contoso.com (CONTOSO). The main page is 'Exports'. The top navigation bar includes 'Add', 'Refresh', 'Run now', 'Enable', 'Disable', and 'Delete'. Below the navigation bar, there's a message: 'How satisfied are you with exports? →'. The 'Scope' dropdown is set to 'Trey Research'. A search bar is also present.

## File partitioning for large datasets

If you have a Microsoft Customer Agreement, Microsoft Partner Agreement, or Enterprise Agreement, you can enable Exports to chunk your file into multiple smaller file partitions to help with data ingestion. When you initially configure your export, set the **File Partitioning** setting to **On**. The setting is **Off** by default.

The screenshot shows the 'File Partitioning' configuration settings. It includes a description: 'Enable partitioning if you have larger datasets and want your exports to be split into multiple files. Please note that if you have partitioning off and it is subsequently turned on, your file schema may change slightly.' A 'Learn more' link is provided. A toggle switch is set to 'On', which is highlighted with a red box.

If you don't have a Microsoft Customer Agreement, Microsoft Partner Agreement, or Enterprise Agreement, then you won't see the **File Partitioning** option.

Partitioning isn't currently supported for resource groups or management group scopes.

## Update existing exports to use file partitioning

If you have existing exports and you want to set up file partitioning, create a new export. File partitioning is only available with the latest Exports version. There may be minor changes to some of the fields in the usage files that get created.

If you enable file partitioning on an existing export, you may see minor changes to the fields in file output. Any changes are due to updates that were made to Exports after you initially set yours up.

## Partitioning output

When file partitioning is enabled, you get a file for each partition of data in the export along with a \_manifest.json file. The manifest contains a summary of the full dataset and information for each file partition in it. Each file partition has headers and contains only a subset of the full dataset. To handle the full dataset, you must ingest each partition of the export.

Here's a \_manifest.json example manifest file.

JSON

```
{
 "manifestVersion": "2021-01-01",
 "dataFormat": "csv",
 "blobCount": 1,
 "byteCount": 160769,
 "dataRowCount": 136,
 "blobs": [
 {
 "blobName": "blobName.csv",
 "byteCount": 160769,
 "dataRowCount": 136,
 "headerRowCount": 1,
 "contentMD5": "md5Hash"
 }
]
}
```

## Export versions

When you create a scheduled export in the Azure portal or with the API, it always runs on the exports version used at creation time. Azure keeps your previously created exports on the same version, unless you update it. Doing so prevents changes in the charges and to CSV fields if the export version is changed. As the export functionality changes over time, field names are sometimes changed and new fields are added.

If you want to use the latest data and fields available, we recommend that you create a new export in the Azure portal. To update an existing export to the latest version, update it in the Azure portal or with the latest Export API version. Updating an existing export might cause you to see minor differences in the fields and charges in files that are produced afterward.

## Verify that data is collected

You can easily verify that your Cost Management data is being collected and view the exported CSV file using Azure Storage Explorer.

In the export list, select the storage account name. On the storage account page, select Open in Explorer. If you see a confirmation box, select Yes to open the file in Azure Storage Explorer.

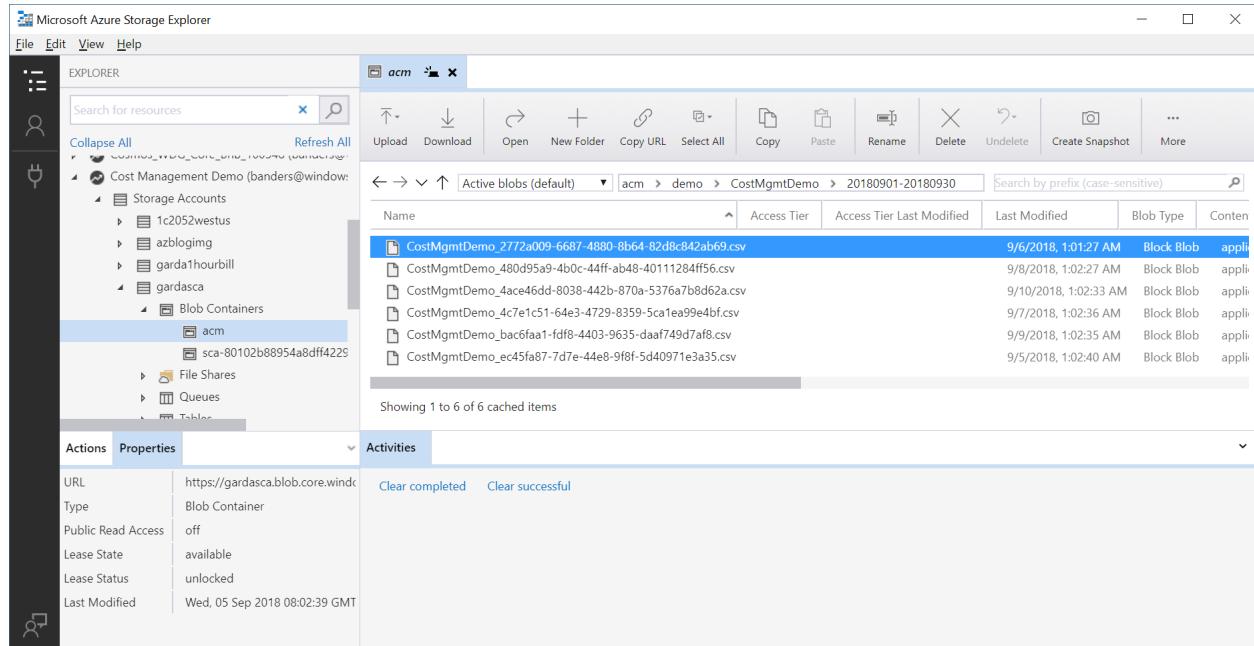
The screenshot shows the Azure Storage Explorer interface. At the top, there's a breadcrumb navigation: Home > Cost Management - Export > gardasca. Below this, the storage account 'gardasca' is listed under 'Storage accounts'. A search bar labeled 'Search (Ctrl+ /)' is present. On the right, there are actions: Open in Explorer, Move, Delete, and Refresh. The main content area displays the following details:

- Resource group (change)**: Garda1HourBill
- Status**: Primary: Available, Secondary: Available
- Location**: North Central US, South Central US
- Subscription (change)**: Cost Management Demo
- Subscription ID**: <SubscriptionID>
- Tags (change)**: Click here to add tags
- Services**:
  - Blobs**: REST-based object storage for unstructured data. A 'Learn more' link is provided.

The left sidebar contains a navigation menu with the following items:

- Overview (selected)
- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Storage Explorer (preview)
- Settings**
  - Access keys
  - CORS
  - Configuration
  - Encryption
  - Shared access signature
  - Firewalls and virtual networks

In Storage Explorer, navigate to the container that you want to open and select the folder corresponding to the current month. A list of CSV files is shown. Select one and then select Open.



The file opens with the program or application that's set to open CSV file extensions. Here's an example in Excel.

The screenshot shows a Microsoft Excel spreadsheet with data from a CSV file. The columns are labeled A through G. The data consists of four rows of information, each containing values for DepartmentName, AccountName, AccountOwnerId, SubscriptionGuid, SubscriptionName, ResourceGroupName, and ResourceLocation. The last row is highlighted in green.

	A	B	C	D	E	F	G
1	DepartmentName	AccountName	AccountOwnerId	SubscriptionGuid	SubscriptionName	ResourceGroupName	ResourceLocation
2	Ama	AAAA	maeptest3@hotmail.com	1caa5a3-2b66-43	Cost Management	Garda1HourBil	usnorthcentral
3	Ama	AAAA	maeptest3@hotmail.com	1caa5a3-2b66-43	Cost Management	MAR-CCM	usnorthcentral
4	Ama	AAAA	maeptest3@hotmail.com	1caa5a3-2b66-43	Cost Management	MAR-CCM	northcentralus

## Download an exported CSV data file

You can also download the exported CSV file in the Azure portal. The following steps explain how to find it from cost analysis.

1. In cost analysis, select **Settings**, and then select **Exports**.
2. In the list of exports, select the storage account for an export.
3. In your storage account, select **Containers**.
4. In list of containers, select the container.
5. Navigate through the directories and storage blobs to the date you want.
6. Select the CSV file and then select **Download**.

Home > ACM\_UX\_DEV | Cost analysis > Cost Management: Cost Management Research | Cost analysis > Exports > treybilling | Containers > costmgmt

**costmgmt**  
Container

Search (Ctrl+F) <>

Upload Change access level Refresh Delete Change tier Acquire lease Break lease

Authentication method: Access key (Switch to Azure AD User Account)  
Location: costmgmt / research / Daily / 20200401-20200430

Search blobs by prefix (case-sensitive)

Name	Modified	Access tier	Blob type	Size	Lease state	Actions
Daily_0aa9876a-e264-4c93-a2b3-24a7d6379b9a.csv	4/3/2020, 6:01:00 AM	Hot (Inferred)	Block blob	108.46 KiB	Available	<a href="#">View/edit</a> <a href="#">Download</a> <a href="#">Properties</a> <a href="#">Edit metadata</a> <a href="#">Generate SAS</a> <a href="#">View snapshots</a> <a href="#">Create snapshot</a> <a href="#">Change tier</a> <a href="#">Acquire lease</a> <a href="#">Break lease</a> <a href="#">Delete</a>
Daily_0bf2189f-41a0-45a3-bc3c-d422a21e934c.csv	4/6/2020, 6:02:09 AM	Hot (Inferred)	Block blob	280.14 KiB	Available	<a href="#">View/edit</a> <a href="#">Download</a> <a href="#">Properties</a> <a href="#">Edit metadata</a> <a href="#">Generate SAS</a> <a href="#">View snapshots</a> <a href="#">Create snapshot</a> <a href="#">Change tier</a> <a href="#">Acquire lease</a> <a href="#">Break lease</a> <a href="#">Delete</a>
Daily_0cb8d46b-d191-4417-9989-5e9ef3c8b665.csv	4/12/2020, 8:01:11 PM	Hot (Inferred)	Block blob	681.5 KiB	Available	<a href="#">View/edit</a> <a href="#">Download</a> <a href="#">Properties</a> <a href="#">Edit metadata</a> <a href="#">Generate SAS</a> <a href="#">View snapshots</a> <a href="#">Create snapshot</a> <a href="#">Change tier</a> <a href="#">Acquire lease</a> <a href="#">Break lease</a> <a href="#">Delete</a>
Daily_227de81-3664-460d-b66c-dd9b1667d568.csv	4/4/2020, 6:01:28 AM	Hot (Inferred)	Block blob	166.04 KiB	Available	<a href="#">View/edit</a> <a href="#">Download</a> <a href="#">Properties</a> <a href="#">Edit metadata</a> <a href="#">Generate SAS</a> <a href="#">View snapshots</a> <a href="#">Create snapshot</a> <a href="#">Change tier</a> <a href="#">Acquire lease</a> <a href="#">Break lease</a> <a href="#">Delete</a>
Daily_6d54f5af-8604-4846-b7bd-b00ec667e26e.csv	4/10/2020, 8:31:11 PM	Hot (Inferred)	Block blob	567.67 KiB	Available	<a href="#">View/edit</a> <a href="#">Download</a> <a href="#">Properties</a> <a href="#">Edit metadata</a> <a href="#">Generate SAS</a> <a href="#">View snapshots</a> <a href="#">Create snapshot</a> <a href="#">Change tier</a> <a href="#">Acquire lease</a> <a href="#">Break lease</a> <a href="#">Delete</a>
Daily_6d62aae2-e55b-48c5-8314-98a2320b82aa.csv	4/13/2020, 8:01:43 PM	Hot (Inferred)	Block blob	738.96 KiB	Available	<a href="#">View/edit</a> <a href="#">Download</a> <a href="#">Properties</a> <a href="#">Edit metadata</a> <a href="#">Generate SAS</a> <a href="#">View snapshots</a> <a href="#">Create snapshot</a> <a href="#">Change tier</a> <a href="#">Acquire lease</a> <a href="#">Break lease</a> <a href="#">Delete</a>
Daily_71e5d265-8c9d-4e41-97af-6f40560494c8.csv	4/5/2020, 6:01:56 AM	Hot (Inferred)	Block blob	270.24 KiB	Available	<a href="#">View/edit</a> <a href="#">Download</a> <a href="#">Properties</a> <a href="#">Edit metadata</a> <a href="#">Generate SAS</a> <a href="#">View snapshots</a> <a href="#">Create snapshot</a> <a href="#">Change tier</a> <a href="#">Acquire lease</a> <a href="#">Break lease</a> <a href="#">Delete</a>
Daily_bc5d94e7-b699-4cf9-b4d3-950d2c1c0718.csv	4/7/2020, 6:02:18 AM	Hot (Inferred)	Block blob	339.73 KiB	Available	<a href="#">View/edit</a> <a href="#">Download</a> <a href="#">Properties</a> <a href="#">Edit metadata</a> <a href="#">Generate SAS</a> <a href="#">View snapshots</a> <a href="#">Create snapshot</a> <a href="#">Change tier</a> <a href="#">Acquire lease</a> <a href="#">Break lease</a> <a href="#">Delete</a>
Daily_cc29968f-dec0-4ef8-98e6-eehebc3364e9.csv	4/14/2020, 8:01:17 PM	Hot (Inferred)	Block blob	796.26 KiB	Available	<a href="#">View/edit</a> <a href="#">Download</a> <a href="#">Properties</a> <a href="#">Edit metadata</a> <a href="#">Generate SAS</a> <a href="#">View snapshots</a> <a href="#">Create snapshot</a> <a href="#">Change tier</a> <a href="#">Acquire lease</a> <a href="#">Break lease</a> <a href="#">Delete</a>
Daily_d6435f32-2238-4492-8f75-800fad16ac72.csv	4/8/2020, 12:22:48 PM	Hot (Inferred)	Block blob	391.2 KiB	Available	<a href="#">View/edit</a> <a href="#">Download</a> <a href="#">Properties</a> <a href="#">Edit metadata</a> <a href="#">Generate SAS</a> <a href="#">View snapshots</a> <a href="#">Create snapshot</a> <a href="#">Change tier</a> <a href="#">Acquire lease</a> <a href="#">Break lease</a> <a href="#">Delete</a>
Daily_d6710132-d4a3-45f7-977b-f7d198318f4f.csv	4/1/2020, 6:06:30 AM	Hot (Inferred)	Block blob	355.8 KiB	Available	<a href="#">View/edit</a> <a href="#">Download</a> <a href="#">Properties</a> <a href="#">Edit metadata</a> <a href="#">Generate SAS</a> <a href="#">View snapshots</a> <a href="#">Create snapshot</a> <a href="#">Change tier</a> <a href="#">Acquire lease</a> <a href="#">Break lease</a> <a href="#">Delete</a>
Daily_d6f80714-8e52-4ec5-8289-cecd49b0a4d.csv	4/2/2020, 6:06:04 AM	Hot (Inferred)	Block blob	51.71 KiB	Available	<a href="#">View/edit</a> <a href="#">Download</a> <a href="#">Properties</a> <a href="#">Edit metadata</a> <a href="#">Generate SAS</a> <a href="#">View snapshots</a> <a href="#">Create snapshot</a> <a href="#">Change tier</a> <a href="#">Acquire lease</a> <a href="#">Break lease</a> <a href="#">Delete</a>
Daily_ecdb1963-ba20-4d3b-840d-9b59b9f4427b.csv	4/11/2020, 8:01:01 PM	Hot (Inferred)	Block blob	624.58 KiB	Available	<a href="#">View/edit</a> <a href="#">Download</a> <a href="#">Properties</a> <a href="#">Edit metadata</a> <a href="#">Generate SAS</a> <a href="#">View snapshots</a> <a href="#">Create snapshot</a> <a href="#">Change tier</a> <a href="#">Acquire lease</a> <a href="#">Break lease</a> <a href="#">Delete</a>
Daily_ee82af5b-159d-44f7-87da-6f57c2dd249a.csv	4/9/2020, 8:01:16 PM	Hot (Inferred)	Block blob	508.78 KiB	Available	<a href="#">View/edit</a> <a href="#">Download</a> <a href="#">Properties</a> <a href="#">Edit metadata</a> <a href="#">Generate SAS</a> <a href="#">View snapshots</a> <a href="#">Create snapshot</a> <a href="#">Change tier</a> <a href="#">Acquire lease</a> <a href="#">Break lease</a> <a href="#">Delete</a>

## View export run history

You can view the run history of your scheduled export by selecting an individual export in the exports list page. The exports list page also provides you with quick access to view the run time of your previous exports and the next time and export will run. Here's an example showing the run history.

Exports

Add Refresh Run now Enable Disable Delete Help

How satisfied are you with exports? →

Scope : Contoso (Demo) Search to filter items...

Name	Schedule status	Last run	Next run	Frequency	Storage account
daily1	Active	8/5/2020, 2:03 AM PDT	8/6/2020, 1:50 AM PDT	Daily	acmtestdiag
myexport	Active	8/5/2020, 2:03 AM PDT	8/6/2020, 2:02 AM PDT	Daily	azurecostdata
coeslalomtest	Active	8/5/2020, 2:02 AM PDT	8/6/2020, 2:01 AM PDT	Daily	usededetailsacmdemo
aopscoetest1	Active	8/5/2020, 2:03 AM PDT	8/6/2020, 2:02 AM PDT	Daily	cloudynacmpipeline
Daily	Active	5/20/2020, 1:24 PM PDT	---	Daily	treybilling
Daily2	Active	5/20/2020, 1:24 PM PDT	---	Daily	treybilling
DailyMTD	Inactive	5/20/2020, 1:24 PM PDT	---	Daily	westusacmexport1

Select an export to view its run history.

**coeslalomtest**

Exports

Run now | Disable | Delete | Edit | Refresh

Scope	:	Contoso (Demo) (BillingAccount)	Storage account	:	usagedetailsacmdemo
Metric	:	Actual cost	Storage account subscrip...	:	StorageAccountSubscriptionID
Frequency	:	Daily	Storage container	:	consumptionusage
Export start date	:	9/5/2019, 11:59 AM PDT	Storage directory	:	daily
Schedule status	:	Active			

Run history

Execution time	Execution status
Aug 05, 2020, 02:02 AM	<span style="color: green;">✓</span> Succeeded
Aug 04, 2020, 02:04 AM	<span style="color: green;">✓</span> Succeeded
Aug 03, 2020, 02:02 AM	<span style="color: green;">✓</span> Succeeded
Aug 02, 2020, 02:03 AM	<span style="color: green;">✓</span> Succeeded
Aug 01, 2020, 02:02 AM	<span style="color: red;">!</span> Failed ⓘ
Jul 31, 2020, 02:04 AM	<span style="color: green;">✓</span> Succeeded
Jul 30, 2020, 02:03 AM	<span style="color: green;">✓</span> Succeeded
Jul 29, 2020, 02:02 AM	<span style="color: green;">✓</span> Succeeded
Jul 28, 2020, 02:04 AM	<span style="color: green;">✓</span> Succeeded
Jul 27, 2020, 02:02 AM	<span style="color: green;">✓</span> Succeeded

## Export runs twice a day for the first five days of the month

If you've created a daily export, you'll have two runs per day for the first five days of each month. One run executes and creates a file with the current month's cost data. It's the run that's available for you to see in the run history. A second run also executes to create a file with all the costs from the prior month. The second run isn't currently visible in the run history. Azure executes the second run to ensure that your latest file for the past month contains all charges exactly as seen on your invoice. It runs because there are cases where latent usage and charges are included in the invoice up to 72 hours after the calendar month has closed. To learn more about Cost Management usage data updates, see [Cost and usage data updates and retention](#).

## Access exported data from other systems

One of the purposes of exporting your Cost Management data is to access the data from external systems. You might use a dashboard system or other financial system. Such systems vary widely so showing an example wouldn't be practical. However, you can get started with accessing your data from your applications at [Introduction to Azure Storage](#).

# Next steps

In this tutorial, you learned how to:

- ✓ Create a daily export
- ✓ Verify that data is collected

Advance to the next tutorial to optimize and improve efficiency by identifying idle and underutilized resources.

**Review and act on optimization recommendations**

# Tutorial: Seed a historical cost dataset with the Exports API

Article • 07/17/2022

Large organizations often need to analyze their historical costs going back a year or more. Creating the dataset might be needed for targeted one-time inquiries or to set up reporting dashboards to visualize cost trends over time. In either case, you need a way to get the data reliably so that you can load it into a data store that you can query. After your historical cost dataset is seeded, your data store can then be updated as new costs come in so that your reporting is kept up to date. Historical costs rarely change and if so, you'll be notified. So we recommend that you refresh your historical costs no more than once a month.

In this tutorial, you learn how to:

- ✓ Get a bearer token for your service principal
- ✓ Format the request
- ✓ Execute the requests in one-month chunks

## Prerequisites

You need proper permissions to successfully call the Exports API. We recommend using a Service Principal in automation scenarios.

- To learn more, see [Assign permissions to Cost Management APIs](#).
- To learn more about the specific permissions needed for the Exports API, see [Understand and work with scopes](#).

Additionally, you'll need a way to query the API directly. For this tutorial, we recommend using [PostMan](#).

## Get a bearer token for your service principal

To learn how to get a bearer token with a service principal, see [Acquire an Access token](#).

## Format the request

See the following example request and create your own one-time data Export. The following example request creates a one-month Actual Cost dataset in the specified

Azure storage account. We recommend that you request no more than one month's of data per report. If you have a large dataset every month, we recommend setting `partitionData = true` for your one-time export to split it into multiple files. For more information, see [File partitioning for large datasets](#).

#### HTTP

```
PUT
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{enrollmentId}/providers/Microsoft.CostManagement/exports/{ExportName}?api-version=2021-10-01
```

#### Request Headers

```
Authorization: <YOUR BEARER TOKEN>
Accept: */*
Content-Type: application/json
```

#### Request Body

#### JSON

```
{
 "properties": {
 "definition": {
 "dataset": {
 "granularity": "Daily",
 "grouping": []
 },
 "timePeriod": {
 "from": "2021-09-01T00:00:00.000Z",
 "to": "2021-09-30T00:00:00.000Z"
 },
 "timeframe": "Custom",
 "type": "ActualCost"
 },
 "deliveryInfo": {
 "destination": {
 "container": "{containerName}",
 "rootFolderPath": "{folderName}",
 "resourceId":
 "/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Storage/storageAccounts/{storageAccountName}"
 }
 },
 "format": "Csv",
 "partitionData": false
```

```
}
```

## Create Exports in one-month chunks

We recommend creating one-time data exports in one month chunks. If you want to seed a one-year historical dataset, then you should execute 12 Exports API requests - one for each month. After you've seeded your historical dataset, you can then create a scheduled export to continue populating your cost data in Azure storage as your charges accrue over time.

## Run each Export

Now that you have created the Export for each month, you need to manually run each by calling the [Execute API](#). An example request to the API is below.

HTTP

POST

```
https://management.azure.com/{scope}/providers/Microsoft.CostManagement/exports/{exportName}/run?api-version=2021-10-01
```

## Next steps

In this tutorial, you learned how to:

- ✓ Get a bearer token for your service principal
- ✓ Format the request
- ✓ Execute the requests in one-month chunks

To learn more about cost details, see [ingest cost details data](#).

To learn more about what data is available in the cost details dataset, see [Understand cost details data fields](#).

# Export cost data with an Azure Storage account SAS key

Article • 06/07/2023

The following information applies to Microsoft partners only.

Often, partners don't have their own Azure subscriptions in the tenant that's associated with their own Microsoft Partner Agreement. Partners with a Microsoft Partner Agreement plan who are global admins of their billing account can export and copy cost data into a storage account in a different tenant using a shared access service (SAS) key. In other words, a storage account with a SAS key allows the partner to use a storage account that's outside of their partner agreement to receive exported information. This article helps partners create a SAS key and configure Cost Management exports.

## Requirements

- You must be a partner with a Microsoft Partner Agreement and have customers on the Azure Plan.
- You must be global admin for your partner organization's billing account.
- You must have access to configure a storage account that's in a different tenant of your partner organization. You're responsible for maintaining permissions and data access when you export data to your storage account.
- The storage account must not have a firewall configured.
- The storage account configuration must have the **Permitted scope for copy operations (preview)** option set to **From any storage account**.

## Configure Azure Storage with a SAS key

Get a storage account SAS token or create one using the Azure portal. To create one in the Azure portal, use the following steps. To learn more about SAS keys, see [Grant limited access to data with shared access signatures \(SAS\)](#).

1. Navigate to the storage account in the Azure portal.
  - If your account has access to multiple tenants, switch directories to access the storage account. Select your account in the upper right corner of the Azure portal and then select **Switch directories**.
  - You might need to sign in to the Azure portal with the corresponding tenant account to access the storage account.

2. In the left menu, select **Shared access signature**.

The screenshot shows the Azure Storage account settings for 'ContosoStorage'. The left sidebar lists various options like Overview, Activity log, Tags, etc., with 'Shared access signature' selected. The main pane displays the configuration for generating a SAS token. It includes sections for Allowed services (Blob checked), Allowed resource types (Service, Container, Object checked), Allowed permissions (Read, Write, Delete, List, Add, Create checked), Blob versioning permissions (Enables deletion of versions checked), Start and expiry date/time (Start: 01/27/2021, End: 01/27/2021), Allowed IP addresses (example: 168.1.5.65 or 168.1.5.65-168.1.5.70), Allowed protocols (HTTPS only selected), Preferred routing tier (Basic (default) selected), and a Signing key dropdown set to 'key1'. A large blue button at the bottom right says 'Generate SAS and connection string'.

3. Configure the token with the same settings as identified in the preceding image.

- a. Select **Blob** for *Allowed services*.
  - b. Select **Service**, **Container**, and **Object** for *Allowed resource types*.
  - c. Select **Read**, **Write**, **Delete**, **List**, **Add**, and **Create** for *Allowed permissions*.
  - d. Choose expiration and dates. Make sure to update your export SAS token before it expires. The longer the time period you configure before expiration, the longer your export runs before needing a new SAS token.
4. Select **HTTPS only** for *Allowed protocols*.
  5. Select **Basic** for *Preferred routing tier*.
  6. Select **key1** for *Signing key*. If you rotate or update the key that's used to sign the SAS token, you'll need to regenerate a new SAS token for your export.
  7. Select **Generate SAS and connection string**. The **SAS token** value shown is the token that you need when you configure exports.

# Create a new export with a SAS token

Navigate to **Exports** at the billing account scope and create a new export using the following steps.

1. Select **Create**.
2. Configure the Export details as you would for a normal export. You can configure the export to use an existing directory or container or you can specify a new directory or container and exports will create them for you.
3. When configuring Storage, select **Use a SAS token**.

The screenshot shows the 'New export' configuration page in the Azure Cost Management portal. The 'Storage' section is the focus, with the 'Use SAS Token' radio button selected. Other fields include 'Name' (contoso), 'Metric' (Actual cost), 'Export type' (Daily export of month-to-date costs), and 'Start date' (Wed Jan 27 2021). The 'Storage' section also includes fields for 'Storage account', 'SAS Token', 'Container', and 'Directory', each with a corresponding input field and a search icon.

4. Enter the name of the storage account and paste in your SAS token.
5. Specify an existing container or Directory or identify new ones to be created.
6. Select **Create**.

The SAS token-based export only works while the token remains valid. Reset the token before the current one expires, or your export will stop working. Because the token provides access to your storage account, protect the token as carefully as you would any other sensitive information. You're responsible to maintain permissions and data access when your export data to your storage account.

## Troubleshoot exports using SAS tokens

The following are common issues that might happen when you configure or use SAS token-based exports.

- You don't see the SAS key option in the Azure portal.
  - Verify that you're a partner that has a Microsoft Partner Agreement and that you have global admin permission to the billing account. They're the only people who can export with a SAS key.
- You get the following error message when trying to configure your export:

**Please ensure the SAS token is valid for blob service, is valid for container and object resource types, and has permissions: add create read write delete.**  
**(Storage service error code: AuthorizationResourceTypeMismatch)**

  - Make sure that you're configuring and generating the SAS key correctly in Azure Storage.
- You can't see the full SAS key after you create an export.
  - Not seeing the key is expected behavior. After the SAS Export is configured, the key is hidden for security reasons.
- You can't access the storage account from the tenant where the export is configured.
  - It's expected behavior. If the storage account is in another tenant, you need to navigate to that tenant first in the Azure portal to find the storage account.
- Your export fails because of a SAS token-related error.
  - Your export works only while the SAS token remains valid. Create a new key and run the export.

## Next steps

- For more information about exporting Cost Management data, see [Create and export data](#).
- For information about exporting large amounts of usage data, see [Retrieve large datasets with exports](#).

# Manage costs with automation

Article • 03/26/2023

You can use Cost Management automation to build a custom set of solutions to retrieve and manage cost data. This article covers common scenarios for Cost Management automation and options available based on your situation. If you want to develop using APIs, common API request examples are presented to help accelerate your development process.

## Automate cost data retrieval for offline analysis

You might need to download your Azure cost data to merge it with other datasets. Or you might need to integrate cost data into your own systems. There are different options available depending on the amount of data involved. You must have Cost Management permissions at the appropriate scope to use APIs and tools in any case. For more information, see [Assign access to data](#).

## Suggestions for handling large datasets

If your organization has a large Azure presence across many resources or subscriptions, you'll have a large amount of usage details data. Excel often can't load such large files. In this situation, we recommend the following options:

### Power BI

Power BI is used to ingest and handle large amounts of data. If you're an Enterprise Agreement customer, you can use the Power BI template app to analyze costs for your billing account. The report contains key views used by customers. For more information, see [Analyze Azure costs with the Power BI template app](#).

### Power BI data connector

If you want to analyze your data daily, we recommend using the [Power BI data connector](#) to get data for detailed analysis. Any reports that you create are kept up to date by the connector as more costs accrue.

### Cost Management exports

You might not need to analyze the data daily. If so, consider using Cost Management's [Exports](#) feature to schedule data exports to an Azure Storage account. Then you can load the data into Power BI as needed, or analyze it in Excel if the file is small enough.

Exports are available in the Azure portal or you can configure exports with the [Exports API](#).

## Usage Details API

Consider using the [Usage Details API](#) if you have a small cost data set. Here are recommended best practices:

- If you want to get the latest cost data, we recommend that you query at most once per day. Reports are refreshed every four hours. If you call more frequently, you'll receive identical data.
- Once you download your cost data for historical invoices, the charges won't change unless you're explicitly notified. We recommend caching your cost data in a queryable store on to prevent repeated calls for identical data.
- Chunk your calls into small date ranges to get more manageable files that you can download. For example, we recommend chunking by day or by week if you have large Azure usage files month-to-month.
- If you have scopes with a large amount of usage data (for example a Billing Account), consider placing multiple calls to child scopes so you get more manageable files that you can download.
- If your dataset is more than 2 GB month-to-month, consider using [exports](#) as a more scalable solution.

## Automate retrieval with Usage Details API

The [Usage Details API](#) provides an easy way to get raw, unaggregated cost data that corresponds to your Azure bill. The API is useful when your organization needs a programmatic data retrieval solution. Consider using the API if you're looking to analyze smaller cost data sets. However, you should use other solutions identified previously if you have larger datasets. The data in Usage Details is provided on a per meter basis, per day. It's used when calculating your monthly bill. The general availability (GA) version of the APIs is `2019-10-01`. Use `2019-04-01-preview` to access the preview version for reservation and Azure Marketplace purchases with the APIs.

If you want to get large amounts of exported data regularly, see [Retrieve large cost datasets recurrently with exports](#).

## Usage Details API suggestions

### Request schedule

We recommend that you make *no more than one request* to the Usage Details API per day. For more information about how often cost data is refreshed and how rounding is handled, see [Understand cost management data](#).

## Target top-level scopes without filtering

Use the API to get all the data you need at the highest-level scope available. Wait until all needed data is ingested before doing any filtering, grouping, or aggregated analysis. The API is optimized specifically to provide large amounts of unaggregated raw cost data. To learn more about scopes available in Cost Management, see [Understand and work with scopes](#). Once you've downloaded the needed data for a scope, use Excel to analyze data further with filters and pivot tables.

## Notes about pricing

If you want to reconcile usage and charges with your price sheet or invoice, note the following information.

Price Sheet price behavior - The prices shown on the price sheet are the prices that you receive from Azure. They're scaled to a specific unit of measure. Unfortunately, the unit of measure doesn't always align to the unit of measure at which the actual resource usage and charges are emitted.

Usage Details price behavior - Usage files show scaled information that may not match precisely with the price sheet. Specifically:

- Unit Price - The price is scaled to match the unit of measure at which the charges are actually emitted by Azure resources. If scaling occurs, then the price won't match the price seen in the Price Sheet.
- Unit of Measure - Represents the unit of measure at which charges are actually emitted by Azure resources.
- Effective Price / Resource Rate - The price represents the actual rate that you end up paying per unit, after discounts are taken into account. It's the price that should be used with the Quantity to do Price \* Quantity calculations to reconcile charges. The price takes into account the following scenarios and the scaled unit price that's also present in the files. As a result, it might differ from the scaled unit price.
  - Tiered pricing - For example: \$10 for the first 100 units, \$8 for the next 100 units.
  - Included quantity - For example: The first 100 units are free and then \$10 per unit.
  - Reservations

- Rounding that occurs during calculation – Rounding takes into account the consumed quantity, tiered/included quantity pricing, and the scaled unit price.

## A single resource might have multiple records for a single day

Azure resource providers emit usage and charges to the billing system and populate the `Additional Info` field of the usage records. Occasionally, resource providers might emit usage for a given day and stamp the records with different datacenters in the `Additional Info` field of the usage records. It can cause multiple records for a meter/resource to be present in your usage file for a single day. In that situation, you aren't overcharged. The multiple records represent the full cost of the meter for the resource on that day.

## Example Usage Details API requests

The following example requests are used by Microsoft customers to address common scenarios that you might come across.

### Get Usage Details for a scope during specific date range

The data that's returned by the request corresponds to the date when the usage was received by the billing system. It might include costs from multiple invoices. The call to use varies by your subscription type.

For legacy customers with an Enterprise Agreement (EA) or a pay-as-you-go subscription, use the following call:

HTTP

GET

```
https://management.azure.com/{scope}/providers/Microsoft.Consumption/usageDetails?$filter=properties%2FusageStart%20ge%20'2020-02-01'%20and%20properties%2FusageEnd%20le%20'2020-02-29'&$top=1000&api-version=2019-10-01
```

For modern customers with a Microsoft Customer Agreement, use the following call:

HTTP

GET

```
https://management.azure.com/{scope}/providers/Microsoft.Consumption/usageDe
```

```
tails?startDate=2020-08-01&endDate=2020-08-05&$top=1000&api-version=2019-10-01
```

### ⓘ Note

The `$filter` parameter isn't supported by Microsoft Customer Agreements.

## Get amortized cost details

If you need actual costs to show purchases as they're accrued, change the *metric* to `ActualCost` in the following request. To use amortized and actual costs, you must use the `2019-04-01-preview` version. The current API version works the same as the `2019-10-01` version, except for the new type/metric attribute and changed property names. If you have a Microsoft Customer Agreement, your filters are `startDate` and `endDate` in the following example.

HTTP

GET

```
https://management.azure.com/{scope}/providers/Microsoft.Consumption/usageDetails?metric=AmortizedCost$filter=properties/usageStart+ge+'2019-04-01'+AND+properties/usageEnd+le+'2019-04-30'&api-version=2019-04-01-preview
```

## Automate alerts and actions with budgets

There are two critical components to maximizing the value of your investment in the cloud. One is automatic budget creation. The other is configuring cost-based orchestration in response to budget alerts. There are different ways to automate budget creation. Various alert responses happen when your configured alert thresholds are exceeded.

The following sections cover available options and provide sample API requests to get you started with budget automation.

## How costs are evaluated against your budget threshold

Your costs are evaluated against your budget threshold once per day. When you create a new budget or at your budget reset day, the costs compared to the threshold will be zero/null because the evaluation might not have occurred.

When Azure detects that your costs have crossed the threshold, a notification is triggered within the hour of the detecting period.

## View your current cost

To view your current costs, you need to make a GET call using the [Query API](#).

A GET call to the Budgets API won't return the current costs shown in Cost Analysis. Instead, the call returns your last evaluated cost.

## Automate budget creation

You can automate budget creation using the [Budgets API](#). You can also create a budget with a [budget template](#). Templates are an easy way for you to standardize Azure deployments while ensuring cost control is properly configured and enforced.

## Supported locales for budget alert emails

With budgets, you're alerted when costs cross a set threshold. You can set up to five email recipients per budget. Recipients receive the email alerts within 24 hours of crossing the budget threshold. However, your recipient might need to receive an email in a different language. You can use the following language culture codes with the Budgets API. Set the culture code with the `locale` parameter similar to the following example.

JSON

```
{
 "eTag": "\"1d681a8fc67f77a\"",
 "properties": {
 "timePeriod": {
 "startDate": "2020-07-24T00:00:00Z",
 "endDate": "2022-07-23T00:00:00Z"
 },
 "timeGrain": "BillingMonth",
 "amount": 1,
 "currentSpend": {
 "amount": 0,
 "unit": "USD"
 },
 "category": "Cost",
 "notifications": {
 "actual_GreaterThan_10_Percent": {
 "enabled": true,
 "operator": "GreaterThan",
 "threshold": 20,
 "timeGrain": "BillingMonth",
 "timePeriod": {
 "endDate": "2022-07-23T00:00:00Z",
 "startDate": "2020-07-24T00:00:00Z"
 }
 }
 }
 }
}
```

```

 "locale": "en-us",
 "contactEmails": [
 "user@contoso.com"
],
 "contactRoles": [],
 "contactGroups": [],
 "thresholdType": "Actual"
 }
}
}
}

```

Languages supported by a culture code:

Culture code	Language
en-us	English (United States)
ja-jp	Japanese (Japan)
zh-cn	Chinese (Simplified, China)
de-de	German (Germany)
es-es	Spanish (Spain, International)
fr-fr	French (France)
it-it	Italian (Italy)
ko-kr	Korean (Korea)
pt-br	Portuguese (Brazil)
ru-ru	Russian (Russia)
zh-tw	Chinese (Traditional, Taiwan)
cs-cz	Czech (Czech Republic)
pl-pl	Polish (Poland)
tr-tr	Turkish (Türkiye)
da-dk	Danish (Denmark)
en-gb	English (United Kingdom)
hu-hu	Hungarian (Hungary)
nb-no	Norwegian Bokmal (Norway)
nl-nl	Dutch (Netherlands)

Culture code	Language
pt-pt	Portuguese (Portugal)
sv-se	Swedish (Sweden)

## Common Budgets API configurations

There are many ways to configure a budget in your Azure environment. Consider your scenario first and then identify the configuration options that enable it. Review the following options:

- **Time Grain** - Represents the recurring period your budget uses to accrue and evaluate costs. The most common options are Monthly, Quarterly, and Annual.
- **Time Period** - Represents how long your budget is valid. The budget actively monitors and alerts you only while it remains valid.
- **Notifications**
  - Contact Emails – The email addresses receive alerts when a budget accrues costs and exceeds defined thresholds.
  - Contact Roles - All users who have a matching Azure role on the given scope receive email alerts with this option. For example, Subscription Owners could receive an alert for a budget created at the subscription scope.
  - Contact Groups - The budget calls the configured action groups when an alert threshold is exceeded.
- **Cost dimension filters** - The same filtering you can do in Cost Analysis or the Query API can also be done on your budget. Use this filter to reduce the range of costs that you're monitoring with the budget.

After you've identified the budget creation options that meet your needs, create the budget using the API. The example below helps get you started with a common budget configuration.

### Create a budget filtered to multiple resources and tags

Request URL: `PUT https://management.azure.com/subscriptions/{SubscriptionId}/providers/Microsoft.Consumption/budgets/{BudgetName}/?api-version=2019-10-01`

JSON

```
{
 "eTag": "\"1d34d016a593709\"",
 "properties": {
 "category": "Cost",
 "amount": 100.65,
 "timeGrain": "P1M",
 "timePeriod": "2019-10-01T00:00:00Z/2020-09-30T23:59:59Z",
 "filters": [
 {
 "resourceType": "VirtualMachine"
 }
],
 "tags": [
 "tag1"
]
 }
}
```

```
"timeGrain": "Monthly",
"timePeriod": {
 "startDate": "2017-10-01T00:00:00Z",
 "endDate": "2018-10-31T00:00:00Z"
},
"filter": {
 "and": [
 {
 "dimensions": {
 "name": "ResourceId",
 "operator": "In",
 "values": [
 "/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/virtualMachines/{meterName}",
 "/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/virtualMachines/{meterName}"
]
 }
 },
 {
 "tags": {
 "name": "category",
 "operator": "In",
 "values": [
 "Dev",
 "Prod"
]
 }
 },
 {
 "tags": {
 "name": "department",
 "operator": "In",
 "values": [
 "engineering",
 "sales"
]
 }
 }
]
},
"notifications": {
 "Actual_GreaterThan_80_Percent": {
 "enabled": true,
 "operator": "GreaterThan",
 "threshold": 80,
 "contactEmails": [
 "user1@contoso.com",
 "user2@contoso.com"
],
 "contactRoles": [
 "Contributor",
 "Reader"
]
 }
}
```

```
],
 "contactGroups": [
 "/subscriptions/{subscriptionID}/resourceGroups/{resourceGroupName}/providers/microsoft.insights/actionGroups/{actionGroupName}"
],
 "thresholdType": "Actual"
}
}
}
```

## Configure cost-based orchestration for budget alerts

You can configure budgets to start automated actions using Azure Action Groups. To learn more about automating actions using budgets, see [Automation with budgets](#).

## Data latency and rate limits

We recommend that you call the APIs no more than once per day. Cost Management data is refreshed every four hours as new usage data is received from Azure resource providers. Calling more frequently doesn't provide more data. Instead, it creates increased load.

## Query API query processing units

In addition to the existing rate limiting processes, the [Query API](#) also limits processing based on the cost of API calls. The cost of an API call is expressed as query processing units (QPU). QPU is a performance currency, like [Cosmos DB RUs](#). They abstract system resources like CPU and memory.

### QPU calculation

Currently, one QPU is deducted for one month of data queried from the allotted quotas. This logic might change without notice.

### QPU factors

The following factor affects the number of QPUs consumed by an API request.

- Date range, as the date range in the request increases, the number of QPUs consumed increases.

Other QPU factors might be added without notice.

## QPU quotas

The following quotas are configured per tenant. Requests are throttled when any of the following quotas are exhausted.

- 12 QPU per 10 seconds
- 60 QPU per 1 min
- 600 QPU per 1 hour

The quotas maybe be changed as needed and more quotas may be added.

## Response headers

You can examine the response headers to track the number of QPUs consumed by an API request and number of QPUs remaining.

`x-ms-ratelimit-microsoft.costmanagement-qpu-retry-after`

Indicates the time to back-off in seconds. When a request is throttled with 429, back off for the time specified in this header before retrying the request.

`x-ms-ratelimit-microsoft.costmanagement-qpu-consumed`

QPUs consumed by an API call.

`x-ms-ratelimit-microsoft.costmanagement-qpu-remaining`

List of remaining quotas.

## Next steps

- [Analyze Azure costs with the Power BI template app.](#)
- [Create and manage exported data](#) with Exports.
- Learn more about the [Usage Details API](#).

# Automation for partners

Article • 07/17/2022

Azure Cost Management is natively available for direct partners who have onboarded their customers to a Microsoft Customer Agreement and have [purchased an Azure Plan](#). Partners and their customers can use Cost Management APIs common tasks. For more information about non-automation scenarios, see [Cost Management for Partners](#).

## Azure Cost Management APIs - Direct and indirect providers

Partners with access to billing scopes in a partner tenant can use the following APIs to view invoiced costs.

APIs at the subscription scope can be called by a partner regardless of the cost policy, as long as they have access to the subscription. Other users with access to the subscription, like the customer or reseller, can call the APIs only after the partner enables the cost policy for the customer tenant.

### To get a list of billing accounts

HTTP

GET

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts?
api-version=2019-10-01-preview
```

### To get a list of customers

HTTP

GET

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{bi
llingAccountName}/customers?api-version=2019-10-01-preview
```

### To get a list of subscriptions

HTTP

GET

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountName}/billingSubscriptions?api-version=2019-10-01-preview
```

## To get a list of invoices for a period of time

HTTP

GET

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountName}/invoices?api-version=2019-10-01-preview&periodStartDate={periodStartDate}&periodEndDate={periodEndDate}
```

The API call returns an array of invoices that has elements similar to the following JSON code.

JSON

```
{ "id": "/providers/Microsoft.Billing/billingAccounts/{billingAccountID}/billingProfiles/{BillingProfileID}/invoices/{InvoiceID}", "name": "{InvoiceID}", "properties": { "amountDue": { "currency": "USD", "value": x.xx }, ... } }
```

Use the preceding returned ID field value and replace it in the following example as the scope to query for usage details.

HTTP

GET

```
https://management.azure.com/{id}/providers/Microsoft.Consumption/UsageDetails?api-version=2019-10-01
```

The example returns the usage records associated with the specific invoice.

## To get the policy for customers to view costs

HTTP

GET

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountName}/customers/{customerID}/policies/default?api-version=2019-10-01-preview
```

## To set the policy for customers to view costs

HTTP

PUT

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountName}/customers/{customerID}/policies/default?api-version=2019-10-01-preview
```

## To get Azure service usage for a billing account

We recommend that you configure an Export for these scenarios. For more information, see [Retrieve large usage datasets with exports](#).

## To download a customer's Azure service usage

We recommend that you configure an Export for this scenario as well. If you need to download the data on demand, however, you can use the [Cost Details API](#). For more information, see [Get small cost datasets on demand](#).

## To get or download the price sheet for consumed Azure services

First, use the following post.

HTTP

POST

```
https://management.azure.com/providers/Microsoft.Billing/BillingAccounts/{billingAccountName}/billingProfiles/{billingProfileID}/pricesheet/default/download?api-version=2019-10-01-preview&format=csv" -verbose
```

Then, call the asynchronous operation property value. For example:

HTTP

GET

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountName}/billingProfiles/{billingProfileID}/pricesheetDownloadOperations/{operation}?sessiontoken=0:11186&api-version=2019-10-01-preview
```

The preceding get call returns the download link containing the price sheet.

## To get aggregated costs

HTTP

POST

```
https://management.azure.com/providers/microsoft.billing/billingAccounts/{billingAccountName}/providers/microsoft.costmanagement/query?api-version=2019-10-01
```

## Create a budget for a partner

HTTP

PUT

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountName}/providers/Microsoft.CostManagement/budgets/partnerworkshopbudget?api-version=2019-10-01
```

## Create a budget for a customer

HTTP

PUT

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountName}/customers/{customerId}/providers/Microsoft.Consumption/budgets/{budgetName}?api-version=2019-10-01
```

## Delete a budget

HTTP

DELETE

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountId}/providers/Microsoft.CostManagement/budgets/{budgetName}?api-version=2019-10-01
```

## Next steps

- Learn more about Cost Management automation at [Cost Management automation overview](#). Automation scenarios.
- Get started with [Azure Cost Management for partners](#).
- Retrieve large usage datasets with [exports](#).

- Understand usage details fields.

# Automation for partners

Article • 07/17/2022

Azure Cost Management is natively available for direct partners who have onboarded their customers to a Microsoft Customer Agreement and have [purchased an Azure Plan](#). Partners and their customers can use Cost Management APIs common tasks. For more information about non-automation scenarios, see [Cost Management for Partners](#).

## Azure Cost Management APIs - Direct and indirect providers

Partners with access to billing scopes in a partner tenant can use the following APIs to view invoiced costs.

APIs at the subscription scope can be called by a partner regardless of the cost policy, as long as they have access to the subscription. Other users with access to the subscription, like the customer or reseller, can call the APIs only after the partner enables the cost policy for the customer tenant.

### To get a list of billing accounts

HTTP

GET

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts?
api-version=2019-10-01-preview
```

### To get a list of customers

HTTP

GET

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{bi
llingAccountName}/customers?api-version=2019-10-01-preview
```

### To get a list of subscriptions

HTTP

GET

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountName}/billingSubscriptions?api-version=2019-10-01-preview
```

## To get a list of invoices for a period of time

HTTP

GET

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountName}/invoices?api-version=2019-10-01-preview&periodStartDate={periodStartDate}&periodEndDate={periodEndDate}
```

The API call returns an array of invoices that has elements similar to the following JSON code.

JSON

```
{ "id": "/providers/Microsoft.Billing/billingAccounts/{billingAccountID}/billingProfiles/{BillingProfileID}/invoices/{InvoiceID}", "name": "{InvoiceID}", "properties": { "amountDue": { "currency": "USD", "value": x.xx }, ... } }
```

Use the preceding returned ID field value and replace it in the following example as the scope to query for usage details.

HTTP

GET

```
https://management.azure.com/{id}/providers/Microsoft.Consumption/UsageDetails?api-version=2019-10-01
```

The example returns the usage records associated with the specific invoice.

## To get the policy for customers to view costs

HTTP

GET

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountName}/customers/{customerID}/policies/default?api-version=2019-10-01-preview
```

## To set the policy for customers to view costs

HTTP

PUT

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountName}/customers/{customerID}/policies/default?api-version=2019-10-01-preview
```

## To get Azure service usage for a billing account

We recommend that you configure an Export for these scenarios. For more information, see [Retrieve large usage datasets with exports](#).

## To download a customer's Azure service usage

We recommend that you configure an Export for this scenario as well. If you need to download the data on demand, however, you can use the [Cost Details API](#). For more information, see [Get small cost datasets on demand](#).

## To get or download the price sheet for consumed Azure services

First, use the following post.

HTTP

POST

```
https://management.azure.com/providers/Microsoft.Billing/BillingAccounts/{billingAccountName}/billingProfiles/{billingProfileID}/pricesheet/default/download?api-version=2019-10-01-preview&format=csv" -verbose
```

Then, call the asynchronous operation property value. For example:

HTTP

GET

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountName}/billingProfiles/{billingProfileID}/pricesheetDownloadOperations/{operation}?sessiontoken=0:11186&api-version=2019-10-01-preview
```

The preceding get call returns the download link containing the price sheet.

## To get aggregated costs

HTTP

POST

```
https://management.azure.com/providers/microsoft.billing/billingAccounts/{billingAccountName}/providers/microsoft.costmanagement/query?api-version=2019-10-01
```

## Create a budget for a partner

HTTP

PUT

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountName}/providers/Microsoft.CostManagement/budgets/partnerworkshopbudget?api-version=2019-10-01
```

## Create a budget for a customer

HTTP

PUT

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountName}/customers/{customerId}/providers/Microsoft.Consumption/budgets/{budgetName}?api-version=2019-10-01
```

## Delete a budget

HTTP

DELETE

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountId}/providers/Microsoft.CostManagement/budgets/{budgetName}?api-version=2019-10-01
```

## Next steps

- Learn more about Cost Management automation at [Cost Management automation overview](#). Automation scenarios.
- Get started with [Azure Cost Management for partners](#).
- Retrieve large usage datasets with [exports](#).

- Understand usage details fields.

# Assign permissions to Cost Management APIs

Article • 04/13/2023

Before using the Azure Cost Management APIs, you need to properly assign permissions to an Azure service principal. From there you can use the service principal identity to call the APIs.

## Permissions configuration checklist

- Get familiar with the [Azure Resource Manager REST APIs](#).
- Determine which Cost Management APIs you want to use. For more information about available APIs, see [Cost Management automation overview](#).
- Configure service authorization and authentication for the Azure Resource Manager APIs.
  - If you're not already using Azure Resource Manager APIs, [register your client app with Azure AD](#). Registration creates a service principal for you to use to call the APIs.
  - Assign the service principal access to the scopes needed, as outlined below.
  - Update any programming code to use [Azure AD authentication](#) with your service principal.

## Assign service principal access to Azure Resource Manager APIs

After you create a service principal to programmatically call the Azure Resource Manager APIs, you need to assign it the proper permissions to authorize against and execute requests in Azure Resource Manager. There are two permission frameworks for different scenarios.

## Azure billing hierarchy access

If you have an Azure Enterprise Agreement or a Microsoft Customer Agreement, you can configure service principal access to Cost Management data in your billing account. To learn more about the billing hierarchies available and what permissions are needed to call each API in Azure Cost Management, see [Understand and work with scopes](#).

- Enterprise Agreements - To assign service principal permissions to your enterprise billing account, departments, or enrollment account scopes, see [Assign roles to Azure Enterprise Agreement service principal names](#).
- Microsoft Customer Agreements - To assign service principal permissions to your Microsoft Customer Agreement billing account, billing profile, invoice section or customer scopes, see [Manage billing roles in the Azure portal](#). Configure the permission to your service principal in the portal as you would a normal user. If you want to automate permissions assignment, see the [Billing Role Assignments API](#).

## Azure role-based access control

Service principal support extends to Azure-specific scopes, like management groups, subscriptions, and resource groups. You can assign service principal permissions to these scopes directly [in the Azure portal](#) or by using [Azure PowerShell](#).

## Next steps

- Learn more about Cost Management automation at [Cost Management automation overview](#).

# Assign roles to Azure Enterprise Agreement service principal names

Article • 05/31/2023

You can manage your Enterprise Agreement (EA) enrollment in the [Azure Enterprise portal](#). Direct Enterprise customer can now manage Enterprise Agreement(EA) enrollment in [Azure portal](#). You can create different roles to manage your organization, view costs, and create subscriptions. This article helps you automate some of those tasks by using Azure PowerShell and REST APIs with Azure service principal names (SPNs).

## ⓘ Note

If you have multiple EA billing accounts in your organization, you must grant the EA roles to Azure SPNs individually in each EA billing account.

Before you begin, ensure that you're familiar with the following articles:

- [Enterprise agreement roles](#)
- [Sign in with Azure PowerShell](#)
- [How to call REST APIs with Postman](#)

## Create and authenticate your service principal

To automate EA actions by using an SPN, you need to create an Azure Active Directory (Azure AD) application. It can authenticate in an automated manner.

Follow the steps in these articles to create and authenticate your service principal.

- [Create a service principal](#)
- [Get tenant and app ID values for signing in](#)

Here's an example of the application registration page.

### Register an application

...

**⚠** If you are building an application for external users that will be distributed by Microsoft, you must register as a first party application to meet all security, privacy, and compliance policies. [Read our decision guide](#)

\* Name  
The user-facing display name for this application (this can be changed later).  
 ✓

Supported account types  
Who can use this application or access this API?  
 Accounts in this organizational directory only (Microsoft only - Single tenant)  
 Accounts in any organizational directory (Any Azure AD directory - Multitenant)  
 Accounts in any organizational directory (Any Azure AD directory - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)  
 Personal Microsoft accounts only  
[Help me choose...](#) 

## Find your SPN and tenant ID

You also need the object ID of the SPN and the tenant ID of the app. You need this information for permission assignment operations later in this article. All applications are registered in Azure AD in the tenant. Two types of objects get created when the app registration is completed:

- Application object - The application object ID is what you see under App Registrations in Azure AD. The object ID should *not* be used to grant any EA roles.
- Service Principal object - The Service Principal object is what you see in the Enterprise Registration window in Azure AD. The object ID is used to grant EA roles to the SPN.

1. Open Azure Active Directory, and then select **Enterprise applications**.

2. Find your app in the list.

The screenshot shows the 'Enterprise applications | All applications' page in Azure Active Directory. The left sidebar has sections for Overview, Manage (selected), Security, Conditional Access, and Consent and permissions. The main area shows a search bar with 'example-spn' entered. Below it, there are two items in the list:

Name	Homepage URL
 example-spn	
 Power BI Service	

3. Select the app to find the application ID and object ID:

The screenshot shows the 'example-spn | Overview' page in Azure Active Directory. The left sidebar has sections for Overview (selected), Deployment Plan, Manage (selected), Properties (selected), Owners, and Roles and administrators (Preview). The main area is titled 'Properties' and shows the following fields:

Name	Value
Name	example-spn
Application ID	73a38cbe-a444-448e-b6c9-...
Object ID	90ff1aad-fedd-4b4f-b7cf-6...

4. Go to the Microsoft Azure AD **Overview** page to find the tenant ID.

The screenshot shows the Microsoft Azure Active Directory Overview page. On the left, there's a sidebar with links like 'Overview', 'Preview hub', 'Diagnose and solve problems', 'Manage' (with sub-links for 'Users', 'Groups', 'External Identities', 'Roles and administrators', and 'Administrative units'), and 'My feed'. The main area has tabs for 'Overview', 'Monitoring', and 'Tutorials'. Below that is a search bar with the placeholder 'Search your tenant'. Under 'Basic information', it shows the tenant details: Name (Microsoft), Tenant ID (11111111-1111-1111-1111-111111111111, highlighted with a red box), Primary domain (microsoft.onmicrosoft.com), License (Azure AD Premium P2), and a 'View' button. To the right, there are buttons for 'Users', 'Groups', 'Applications', and 'Devices', each with a 'View' link. A magnifying glass icon is also present.

#### i Note

Your tenant ID might be called a principal ID, SPN, or object ID in other locations. The value of your Azure AD tenant ID looks like a GUID with the following format: 11111111-1111-1111-1111-111111111111.

## Permissions that can be assigned to the SPN

Later in this article, you'll give permission to the Azure AD app to act by using an EA role. You can assign only the following roles to the SPN, and you need the role definition ID, exactly as shown.

Role	Actions allowed	Role definition ID
EnrollmentReader	Enrollment readers can view data at the enrollment, department, and account scopes. The data contains charges for all of the subscriptions under the scopes, including across tenants. Can view the Azure Prepayment (previously called monetary commitment) balance associated with the enrollment.	24f8edb6-1668-4659-b5e2-40bb5f3a7d7e
EA purchaser	Purchase reservation orders and view reservation transactions. It has all the permissions of EnrollmentReader, which will in turn have all the permissions of DepartmentReader. It can view usage and charges across all accounts and subscriptions. Can view the Azure Prepayment (previously called monetary commitment) balance associated with the enrollment.	da6647fb-7651-49ee-be91-c43c4877f0c4
DepartmentReader	Download the usage details for the department they administer. Can view the usage and charges associated with their department.	db609904-a47f-4794-9be8-9bd86fbffd8a
SubscriptionCreator	Create new subscriptions in the given scope of Account.	a0bcee42-bf30-4d1b-926a-48d21664ef71

- An EnrollmentReader role can be assigned to an SPN only by a user who has an enrollment writer role. The EnrollmentReader role assigned to an SPN isn't shown in the EA portal. It's created by programmatic means and is only for programmatic use.
- A DepartmentReader role can be assigned to an SPN only by a user who has an enrollment writer or department writer role.
- A SubscriptionCreator role can be assigned to an SPN only by a user who is the owner of the enrollment account (EA administrator). The role isn't shown in the EA portal. It's created by programmatic means and is only for programmatic use.
- The EA purchaser role isn't shown in the EA portal. It's created by programmatic means and is only for programmatic use.

When you grant an EA role to an SPN, you must use the `billingRoleAssignmentName` required property. The parameter is a unique GUID that you must provide. You can generate a GUID using the [New-Guid](#) PowerShell command. You can also use the [Online GUID / UUID Generator](#) website to generate a unique GUID.

An SPN can have only one role.

## Assign enrollment account role permission to the SPN

1. Read the [Role Assignments - Put](#) REST API article. While you read the article, select Try it to get started by using the SPN.

# Role Assignments - Put

Service: Billing

API Version: 2019-10-01-preview

Create or update a billing role assignment.

The screenshot shows the Azure API Management interface for the 'Role Assignments - Put' endpoint. At the top, there's a bar with 'HTTP' and a 'Try It' button, which is highlighted with a red box. Below this is a code snippet: 'PUT https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountName}/billingRoleAssignments/{billingRoleAssignmentName} HTTP/1.1'.

2. Use your account credentials to sign in to the tenant with the enrollment access that you want to assign.

3. Provide the following parameters as part of the API request.

- **billingAccountName**: This parameter is the **Billing account ID**. You can find it in the Azure portal on the [Cost Management + Billing](#) overview page.

The screenshot shows the 'Cost Management + Billing | Overview' page in the Azure portal. On the left, there's a sidebar with 'Overview', 'Access control (IAM)', and 'Billing scopes'. In the main area, under 'Essentials', the 'Billing account ID' is listed as '11111111', which is highlighted with a red box. To the right, there's a 'Charges' section showing 'Month-to-date charges' at '\$12,933.42 USD' and 'Last month charges' at '\$97,687.20 usd'.

- **billingRoleAssignmentName**: This parameter is a unique GUID that you need to provide. You can generate a GUID using the [New-Guid](#) PowerShell command. You can also use the [Online GUID / UUID Generator](#) website to generate a unique GUID.
- **api-version**: Use the 2019-10-01-preview version. Use the sample request body at [Role Assignments - Put - Examples](#).

The request body has JSON code with three parameters that you need to use.

Parameter	Where to find it
<code>properties.principalId</code>	It is the value of Object ID. See <a href="#">Find your SPN and tenant ID</a> .
<code>properties.principalTenantId</code>	See <a href="#">Find your SPN and tenant ID</a> .
<code>properties.roleDefinitionId</code>	/providers/Microsoft.Billing/billingAccounts/{BillingAccountName}/billingRoleDefinitions/24f8edb6-1668-4659-b5e2-40bb5f3a7d7e

The billing account name is the same parameter that you used in the API parameters. It's the enrollment ID that you see in the EA portal and Azure portal.

Notice that `24f8edb6-1668-4659-b5e2-40bb5f3a7d7e` is a billing role definition ID for an EnrollmentReader.

4. Select Run to start the command.

## Request URL

PUT https://management.azure.com/providers/Microsoft.Billing/billingAccounts/1111111/billingRoleAssignment?api-version=2019-10-01-preview

## Parameters

billingAccountName\* 1111111

billingRoleAssignmentName\* 11111111-1111-1111-1111-11111111

api-version\* 2019-10-01-preview

+ name value

## Headers

Content-Type\* application/json

+ name value

## Body

```
{
 "properties": {
 "principalId": "11111111-1111-1111-1111-111111111111",
 "principalTenantId": "11111111-1111-1111-1111-111111111111",
 "roleDefinitionId": "/providers/Microsoft.Billing/billingAccounts/11111111/billingRoleDefinitions/9f1983cb-2574-400c-87e9-34cf8e2280db"
 }
}
```

## Request Preview

HTTP

Copy

```
PUT https://management.azure.com/providers/Microsoft.Billing/billingAccounts/1111111/billingRoleAssignment
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Im5PbzNaRHJPRFhFSzFqS1d0WHNsSFJfS1hFZyIs
Content-type: application/json
```

Run ▶



A 200 OK response shows that the SPN was successfully added.

Now you can use the SPN to automatically access EA APIs. The SPN has the EnrollmentReader role.

## Assign EA Purchaser role permission to the SPN

For the EA purchaser role, use the same steps for the enrollment reader. Specify the `roleDefinitionId`, using the following example:

```
"/providers/Microsoft.Billing/billingAccounts/1111111/billingRoleDefinitions/ da6647fb-7651-49ee-be91-c43c4877f0c4"
```

## Assign the department reader role to the SPN

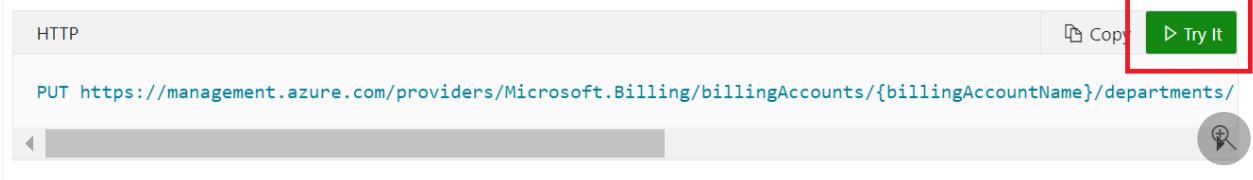
1. Read the [Enrollment Department Role Assignments - Put](#) REST API article. While you read the article, select Try it.

### Enrollment Department Role Assignments - Put

Service: Billing

API Version: 2019-10-01-preview

Create or update a billing role assignment.



2. Use your account credentials to sign in to the tenant with the enrollment access that you want to assign.

3. Provide the following parameters as part of the API request.

- `billingAccountName`: This parameter is the **Billing account ID**. You can find it in the Azure portal on the **Cost Management + Billing** overview page.

The screenshot shows the 'Cost Management + Billing | Overview' page in the Azure portal. At the top, there's a search bar and navigation links for 'EA Portal' and 'Help'. On the left, there's a sidebar with 'Overview', 'Access control (IAM)', and 'Billing scopes'. In the main area, under 'Essentials', it shows 'Billing account ID : 1111111' (which is highlighted with a red box) and 'Billing country/region : US'. On the right, there's a 'Charges' section showing 'Month-to-date charges' at \$12,933.42 USD (Breakdown) and 'Last month charges' at \$97,687.20 USD.

- `billingRoleAssignmentName`: This parameter is a unique GUID that you need to provide. You can generate a GUID using the [New-Guid](#) PowerShell command. You can also use the [Online GUID / UUID Generator](#) website to generate a unique GUID.
- `departmentName`: This parameter is the department ID. You can see department IDs in the Azure portal on the **Cost Management + Billing > Departments** page.

For this example, we used the ACE department. The ID for the example is `84819`.

Name	ID
ACE	84819
ACM	84820
ACS	113247
ADO	105920
AGCE-E2E	106137
AOPS	84821
API-Coder	119404
ASB_Department	121928
ASMS	84822
ASMS 001	97180
ASMS Testing	123022

- **api-version:** Use the 2019-10-01-preview version. Use the sample at [Enrollment Department Role Assignments - Put](#).

The request body has JSON code with three parameters that you need to use.

Parameter	Where to find it
<code>properties.principalId</code>	It is the value of Object ID. See <a href="#">Find your SPN and tenant ID</a> .
<code>properties.principalTenantId</code>	See <a href="#">Find your SPN and tenant ID</a> .
<code>properties.roleDefinitionId</code>	<code>/providers/Microsoft.Billing/billingAccounts/{BillingAccountName}/billingRoleDefinitions/db609904-a47f-4794-9be8-9bd86fbffd8a</code>

The billing account name is the same parameter that you used in the API parameters. It's the enrollment ID that you see in the EA portal and Azure portal.

The billing role definition ID of `db609904-a47f-4794-9be8-9bd86fbffd8a` is for a department reader.

4. Select **Run** to start the command.

## Request URL

PUT https://management.azure.com/providers/Microsoft.Billing/billingAccounts/1111111/departments/84819/billingRoleAssignmentName\*

## Parameters

billingAccountName\* 1111111

billingRoleAssignmentName\* 11111111-1111-1111-1111-111111111111

departmentName\* 84819

api-version\* 2019-10-01-preview

+ name value

## Headers

Content-Type\* application/json

+ name value

## Body

```
{
 "properties": {
 "principalId": "11111111-1111-1111-1111-111111111111",
 "principalTenantId": "11111111-1111-1111-111111111111",
 "roleDefinitionId":
 "/providers/Microsoft.Billing/billingAccounts/1111111/departments/84819/billingRoleDefinitions/11111111-1111-1111-111111111111"
 }
}
```

## Request Preview

HTTP

Copy

```
PUT https://management.azure.com/providers/Microsoft.Billing/billingAccounts/1111111/departments/84819/billingRoleAssignmentName
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Im5PbzNaRHJPRFhFSzFqS1doWHNsSFJfs1hFZyIsImtpZCI6Im5PbzNa
Content-type: application/json
```

Run ▶



A `200 OK` response shows that the SPN was successfully added.

Now you can use the SPN to automatically access EA APIs. The SPN has the DepartmentReader role.

## Assign the subscription creator role to the SPN

1. Read the [Enrollment Account Role Assignments - Put](#) article. While you read it, select Try It to assign the subscription creator role to the SPN.

## Enrollment Department Role Assignments - Put

Service: Billing

API Version: 2019-10-01-preview

Create or update a billing role assignment.

The screenshot shows the 'Try It' interface for the 'Enrollment Department Role Assignments - Put' API. It includes an 'HTTP' method dropdown, a URL input field with a placeholder 'PUT https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountName}/departments/{departmentName}', a 'Copy' button, and a large green 'Try It' button with a red border. Below the input fields is a progress bar and a magnifying glass icon.

2. Use your account credentials to sign in to the tenant with the enrollment access that you want to assign.

3. Provide the following parameters as part of the API request. Read the article at [Enrollment Account Role Assignments - Put - URI Parameters](#).

- `billingAccountName`: This parameter is the **Billing account ID**. You can find it in the Azure portal on the [Cost Management + Billing overview](#) page.

The screenshot shows the 'Cost Management + Billing | Overview' page in the Azure portal. The left sidebar lists 'Overview', 'Access control (IAM)', and 'Billing scopes'. Under 'Cost Management', there are links for 'Cost analysis', 'Cost alerts', 'Budgets', and 'Advisor recommendations'. The main area displays the 'Billing account ID' as '1111111' (highlighted with a red box), 'Billing country/region' as 'US', and 'Month-to-date charges' as '12,933.42 USD (Breakdown)'. Below that, 'Last month charges' are shown as '97,687.20 USD'. A search bar, 'EA Portal' link, and 'Help' link are also visible.

- `billingRoleAssignmentName`: This parameter is a unique GUID that you need to provide. You can generate a GUID using the [New-Guid](#) PowerShell command. You can also use the [Online GUID/UUID Generator](#) website to generate a unique GUID.
- `enrollmentAccountName`: This parameter is the account ID. Find the account ID for the account name in the Azure portal on the [Cost Management + Billing](#) page.

For this example, we used the GTM Test Account. The ID is `196987`.

**Cost Management + Billing | Accounts**

Microsoft

Search (Ctrl+/) Export

Overview Access control (IAM) Billing scopes

**Cost Management**

- Cost analysis
- Cost alerts
- Budgets
- Advisor recommendations
- Cloudyn

**Billing**

- Usage + charges
- Credits
- Reservation transactions
- Departments
- Accounts**

gtm

Name	ID	Account owner	Start date
GTM_Test	111111	admin@ACEGTM.on...	10/5/2018
GTM Test Account	196987	admin@GtmAceSign...	11/6/2018

- **api-version:** Use the 2019-10-01-preview version. Use the sample at [Enrollment Department Role Assignments - Put - Examples](#).

The request body has JSON code with three parameters that you need to use.

Parameter	Where to find it
properties.principalId	It is the value of Object ID. See <a href="#">Find your SPN and tenant ID</a> .
properties.principalTenantId	See <a href="#">Find your SPN and tenant ID</a> .
properties.roleDefinitionId	/providers/Microsoft.Billing/billingAccounts/{BillingAccountId}/enrollmentAccounts/{enrollmentAccountId}/billingRoleDefinitions/a0bcee42-bf30-4d1b-926a-48d21664ef71

The billing account name is the same parameter that you used in the API parameters. It's the enrollment ID that you see in the EA portal and the Azure portal.

The billing role definition ID of `a0bcee42-bf30-4d1b-926a-48d21664ef71` is for the subscription creator role.

4. Select **Run** to start the command.

## REST API Try It

Try the REST API with the inputs below.

[Sign out](#)

### Request URL

PUT https://management.azure.com/providers/Microsoft.Billing/billingAccounts/111111/enrollmentAccounts/196987/billingRoleAssignments/11111

### Parameters

billingAccountName\* 111111

billingRoleAssignmentName\* 11111111-1111-1111-1111-111111111111

enrollmentAccountName\* 196987

api-version\* 2019-10-01-preview

[name](#) [value](#) [+](#)

### Headers

Content-Type\* application/json

[name](#) [value](#) [+](#)

### Body

```
{
 "properties": {
 "principalId": "11111111-1111-1111-1111-111111111111",
 "principalTenantId": "11111111-1111-1111-1111-111111111111",
 "roleDefinitionId": "/providers/Microsoft.Billing/billingAccounts/11111111/enrollmentAccounts/196987/billingRoleDefinitions/11111111-1111-1111-111111111111"
 }
}
```

### Request Preview

HTTP

[Copy](#)

```
PUT https://management.azure.com/providers/Microsoft.Billing/billingAccounts/111111/enrollmentAccounts/196987/billingRoleAssignments/11111111-1111-111111111111
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Im5PbzNaRHJPRFhFSzFqS1doWHNsSFJfS1hFZyIsImtpZCI6Im5PbzNaRHJPRFhFSzFqS1doWHNsSFJfS
Content-type: application/json
```



A `200 OK` response shows that the SPN has been successfully added.

Now you can use the SPN to automatically access EA APIs. The SPN has the `SubscriptionCreator` role.

## Verify SPN role assignments

SPN role assignments are not visible in the Azure portal. You can view enrollment account role assignments, including the subscription creator role, with the [Billing Role Assignments - List By Enrollment Account - REST API \(Azure Billing\)](#) API. Use the API to verify that the role assignment was successful.

## Troubleshoot

You must identify and use the Enterprise application object ID where you granted the EA role. If you use the Object ID from some other application, API calls will fail. Verify that you're using the correct Enterprise application object ID.

If you receive the following error when making your API call, then you may be incorrectly using the SPN object ID value located in App Registrations. To resolve this error, ensure you're using the SPN object ID from Enterprise Applications, not App Registrations.

```
The provided principal Tenant Id = xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxx and principal Object Id xxxxxxxx-xxxx-xxxx-xxxx-
xxxxxxxxxxx are not valid
```

## Next steps

Learn more about [Azure EA portal administration](#).

# Ingest cost details data

Article • 05/23/2023

Cost details (formerly referred to as usage details) are the most granular cost records that are available across Microsoft. Cost details records allow you to correlate Azure meter-based charges with the specific resources responsible for the charges so that you can properly reconcile your bill. The data also includes charges associated with New Commerce products like Microsoft 365 and Dynamics 365 that are invoiced along with Azure. Currently, only Partners can purchase New Commerce non-Azure products. To learn more, see [Understand cost management data](#).

This document outlines the main solutions available to you as you work with cost details data. You might need to download your cost data to merge it with other datasets. Or you might need to integrate cost data into your own systems. There are different options available depending on the amount of data involved.

You must have Cost Management permissions at the appropriate scope to use APIs and tools in any case. For more information, see [Assign access to data](#) and [Assign permissions to Cost Management APIs](#).

## How to get cost details

You can use [exports](#) or the [Cost Details](#) report to get cost details programmatically. To learn more about which solutions are best for your scenarios, see [Choose a cost details solution](#).

For Azure portal download instructions, see [How to get your Azure billing invoice and daily usage data](#). If you have a small cost details dataset that you maintain from one month to another, you can open your CSV file in Microsoft Excel or another spreadsheet application.

## Cost details data format

The Azure billing system uses cost details records at the end of the month to generate your bill. Your bill is based on the net charges that were accrued by meter. Cost records contain daily rated usage based on negotiated rates, purchases (for example, reservations, Marketplace fees), and refunds for the specified period. Fees don't include credits, taxes, or other charges or discounts.

The following table shows the charges that are included in your cost details dataset for each account type.

Account type	Azure usage	Marketplace usage	Purchases	Refunds
Enterprise Agreement (EA)	✓	✓	✓	✗
Microsoft Customer Agreement (MCA)	✓	✓	✓	✓
Pay-as-you-go (PAYG)	✓	✓	✗	✗

A single Azure resource often has multiple meters emitting charges. For example, a VM may have both Compute and Networking related meters.

To understand the fields that are available in cost details records, see [Understand cost details fields](#).

To learn more about Marketplace orders (also known as external services), see [Understand your Azure external service charges](#).

## A single resource might have multiple records per day

Azure resource providers emit usage and charges to the billing system and populate the Additional Info field of the usage records. Occasionally, resource providers might emit usage for a given day and stamp the records with different datacenters in the Additional Info field of the cost records. It can cause multiple records for a meter or resource to be present in your cost file for a single day. In that situation, you aren't overcharged. The multiple records represent the full cost of the meter for the resource on that day.

## Pricing behavior in cost details

The cost details file exposes multiple price points today. These are outlined below.

- **PAYGPrice:** It's the list price or on demand price for a given product or service.
  - PAYGPrice is populated only for first party Azure usage charges where `PricingModel` is `OnDemand`. So for EA customers, `PAYGprice` isn't populated when `PricingModel = Reservations, Spot, Marketplace, Or SavingsPlan`.
  - PAYGPrice is the price customers pay if the VM was consumed as a Standard VM, instead of a Spot VM.
- **UnitPrice:** It's the price for a given product or service inclusive of any negotiated discounts on top of the pay-as-you-go price.

- **EffectivePrice** It's the price for a given product or service that represents the actual rate that you end up paying per unit. It's the price that should be used with the Quantity to do Price \* Quantity calculations to reconcile charges. The price takes into account the following scenarios:
  - *Tiered pricing*: For example: \$10 for the first 100 units, \$8 for the next 100 units.
  - *Included quantity*: For example: The first 100 units are free and then \$10 for each unit.
  - *Reservations*: For example, a VM that got a reservation benefit on a given day. In amortized data for reservations, the effective price is the prorated hourly reservation cost. The cost is the total cost of reservation usage by the resource on that day.
  - *Rounding that occurs during calculation*: Rounding takes into account the consumed quantity, tiered/included quantity pricing, and the scaled unit price.
- **Quantity**: This is the number of units used by the given product or service for a given day and is aligned to the unit of measure used in actual resource usage.

If you want to reconcile costs with your price sheet or invoice, note the following information about unit of measure.

**Price Sheet unit of measure behavior** - The prices shown on the price sheet are the prices that you receive from Azure. They're scaled to a specific unit of measure.

**Cost details unit of measure behavior** - The unit of measure associated with the usage quantities and pricing seen in cost details aligns with actual resource usage.

## Example pricing scenarios seen in cost details for a resource

MeterId	Quantity	PAYGPrice	UnitPrice	EffectivePrice	UnitOfMeasure	Notes
00000000-0000-0000-0000-000000000000	24	1	0.8	0.76	1 hour	Manual calculation of the actual charge: multiply $24 * 0.76$ * 1 hour.

## Unexpected charges

If you have charges that you don't recognize, there are several things you can do to help understand why:

- Review the invoice that has charges for the resource
- Review your invoiced charges in Cost analysis
- Find people responsible for the resource and engage with them
- Analyze the audit logs
- Analyze user permissions to the resource's parent scope
- Create an [Azure support request](#) to help identify the charges

For more information, see [Analyze unexpected charges](#).

Azure doesn't log most user actions. Instead, Azure logs resource usage for billing. If you notice a usage spike in the past and you didn't have logging enabled, Azure can't pinpoint the cause. Enable logging for the service that you want to view the increased usage for so that the appropriate technical team can assist you with the issue.

## Next steps

- Learn more about [Choose a cost details solution](#).
- [Create and manage exported data](#) in the Azure portal with Exports.
- [Automate Export creation](#) and ingestion at scale using the API.
- [Understand cost details fields](#).
- Learn how to [Get small cost datasets on demand](#).

# Choose a cost details solution

Article • 07/18/2022

There are multiple ways to work with the cost details dataset (formerly referred to as usage details). If your organization has a large Azure presence across many resources or subscriptions, you'll have a large amount of cost details data. Excel often can't load such large files. In this situation, we recommend the options below.

## Exports

Exports are recurring data dumps to storage that can be configured to run on a custom schedule. We recommend Exports as the solution to ingest cost details data. It's the most scalable for large enterprises. Exports are [configured in the Azure portal](#) or using the [Exports API](#). Review the considerations below for analyzing whether this solution is best for your particular data ingestion workload.

- Exports are most scalable solution for your workloads.
- Can be configured to use file partitioning for bigger datasets.
- Great for establishing and growing a cost dataset that can be integrated with your own queryable data stores.
- Requires access to a storage account that can hold the data.

To learn more about how to properly call the API and ingest cost details at scale, see [Retrieve large datasets with exports](#).

## Cost Details API

The [Cost Details](#) API is the go to solution for on demand download of the cost details dataset. Review the considerations below to analyze whether this solution is best for your particular data ingestion workload.

- Useful for small cost datasets. Exports scale better than the API. The API may not be a good solution if you need to ingest many gigabytes worth of cost data month over month. A GB of cost details data is roughly 1 million rows of data.
- Useful for scenarios when Exports to Azure storage aren't feasible due to security or manageability concerns.

If the Cost Details API is your chosen solution, review the best practices to call the API below.

- If you want to get the latest cost data, we recommend that you query at most once per day. Reports are refreshed every four hours. If you call more frequently, you'll receive identical data.
- Once you download your cost data for historical invoices, the charges won't change unless you're explicitly notified. We recommend caching your cost data in a queryable store on to prevent repeated calls for identical data.
- Chunk your calls into small date ranges to get more manageable files that you can download. For example, we recommend chunking by day or by week if you have large Azure usage files month-to-month.
- If you have scopes with a large amount of usage data (for example a Billing Account), consider placing multiple calls to child scopes so you get more manageable files that you can download.
- If you're bound by rate limits at a lower scope, consider calling a higher scope to download data.
- If your dataset is more than 2 GB month-to-month, consider using [exports](#) as a more scalable solution.

To learn more about how to properly call the [Cost Details API](#), see [Get small usage data sets on demand](#).

The Cost Details API is only available for customers with an Enterprise Agreement or Microsoft Customer Agreement. If you're an MSDN, pay-as-you-go or Visual Studio customer, see [Get usage details for pay-as-you-go subscriptions](#).

## Power BI

Power BI is another solution that's used to work with cost details data. The following Power BI solutions are available:

- Azure Cost Management Template App: - If you're an Enterprise Agreement or Microsoft Customer Agreement customer, you can use the Power BI template app to analyze costs for your billing account. It includes predefined reports that are built on top of the cost details dataset, among others. For more information, see [Analyze Azure costs with the Power BI template app](#).
- Azure Cost Management Connector: - If you want to analyze your data daily, you can use the [Power BI data connector](#) to get data for detailed analysis. Any reports that you create are kept up to date by the connector as more costs accrue.

## Azure portal download

Only [download your usage from the Azure portal](#) if you have a small cost details dataset that is capable of being loaded in Excel. Cost files that are larger than one or 2 GB may take an exceedingly long time to generate on demand from the Azure portal. They'll take longer to transfer over a network to your local computer. We recommend using one of the above solutions if you have a large monthly usage dataset.

## Next steps

- Get an overview of [how to ingest cost data](#).
- [Create and manage exported data](#) in the Azure portal with Exports.
- [Automate Export creation](#) and ingestion at scale using the API.
- [Understand cost details fields](#).
- Learn how to [Get small cost datasets on demand](#).

# Understand cost details fields

Article • 04/04/2023

This document describes the cost details (formerly known as usage details) fields found in files from using [Azure portal download](#), [Exports](#) from Cost Management, or the [Cost Details API](#). For more information about cost details best practices, see [Choose a cost details solution](#).

## Migration to new cost details formats

If you're using an older cost details solution and want to migrate to Exports or the Cost Details API, read the following articles.

- [Migrate from Enterprise Usage Details APIs](#)
- [Migrate from EA to MCA APIs](#)
- [Migrate from Consumption Usage Details API](#)

## List of fields and descriptions

The following table describes the important terms used in the latest version of the cost details file. The list covers pay-as-you-go (also called Microsoft Online Services Program), Enterprise Agreement (EA), Microsoft Customer Agreement (MCA), and Microsoft Partner Agreement (MPA) accounts.

MPA accounts have all MCA terms, in addition to the MPA terms, as described in the following table. To identify what account type you are, see [supported Microsoft Azure offers](#).

Term	Account type	Description
AccountName	EA, pay-as-you-go	Display name of the EA enrollment account or pay-as-you-go billing account.
AccountOwnerId <sup>1</sup>	EA, pay-as-you-go	Unique identifier for the EA enrollment account or pay-as-you-go billing account.
AdditionalInfo <sup>1</sup>	All	Service-specific metadata. For example, an image type for a virtual machine.
BillingAccountId <sup>1</sup>	All	Unique identifier for the root billing account.

<b>Term</b>	<b>Account type</b>	<b>Description</b>
BillingAccountName	All	Name of the billing account.
BillingCurrency	All	Currency associated with the billing account.
BillingPeriod	EA, pay-as-you-go	The billing period of the charge.
BillingPeriodEndDate	All	The end date of the billing period.
BillingPeriodStartDate	All	The start date of the billing period.
BillingProfileId <sup>1</sup>	All	Unique identifier of the EA enrollment, pay-as-you-go subscription, MCA billing profile, or AWS consolidated account.
BillingProfileName	All	Name of the EA enrollment, pay-as-you-go subscription, MCA billing profile, or AWS consolidated account.
ChargeType	All	Indicates whether the charge represents usage ( <b>Usage</b> ), a purchase ( <b>Purchase</b> ), or a refund ( <b>Refund</b> ).
ConsumedService	All	Name of the service the charge is associated with.
CostCenter <sup>1</sup>	EA, MCA	The cost center defined for the subscription for tracking costs (only available in open billing periods for MCA accounts).
Cost	EA, pay-as-you-go	See CostInBillingCurrency.
CostAllocationRuleName	EA, MCA	Name of the Cost Allocation rule that's applicable to the record.
CostInBillingCurrency	MCA	Cost of the charge in the billing currency before credits or taxes.
CostInPricingCurrency	MCA	Cost of the charge in the pricing currency before credits or taxes.
Currency	EA, pay-as-you-go	See <a href="#">BillingCurrency</a> .
CustomerName	MPA	Name of the Azure Active Directory tenant for the customer's subscription.

<b>Term</b>	<b>Account type</b>	<b>Description</b>
CustomerTenantId	MPA	Identifier of the Azure Active Directory tenant of the customer's subscription.
Date <sup>1</sup>	All	The usage or purchase date of the charge.
EffectivePrice	All	Blended unit price for the period. Blended prices average out any fluctuations in the unit price, like graduated tiering, which lowers the price as quantity increases over time.
ExchangeRateDate	MCA	Date the exchange rate was established.
ExchangeRatePricingToBilling	MCA	Exchange rate used to convert the cost in the pricing currency to the billing currency.
Frequency	All	Indicates whether a charge is expected to repeat. Charges can either happen once ( <b>OneTime</b> ), repeat on a monthly or yearly basis ( <b>Recurring</b> ), or be based on usage ( <b>UsageBased</b> ).
InvoiceId	pay-as-you-go, MCA	The unique document ID listed on the invoice PDF.
InvoiceSection	MCA	See <a href="#">InvoiceSectionName</a> .
InvoiceSectionId <sup>1</sup>	EA, MCA	Unique identifier for the EA department or MCA invoice section.
InvoiceSectionName	EA, MCA	Name of the EA department or MCA invoice section.
IsAzureCreditEligible	All	Indicates if the charge is eligible to be paid for using Azure credits (Values: <a href="#">True</a> or <a href="#">False</a> ).
Location	MCA	Normalized location of the resource, if different resource locations are configured for the same regions.
MeterCategory	All	Name of the classification category for the meter. For example, <i>Cloud services</i> and <i>Networking</i> .
MeterId <sup>1</sup>	All	The unique identifier for the meter.
MeterName	All	The name of the meter.
MeterRegion	All	Name of the datacenter location for services priced based on location. See Location.
MeterSubCategory	All	Name of the meter subclassification category.

<b>Term</b>	<b>Account type</b>	<b>Description</b>
OfferId <sup>1</sup>	All	Name of the offer purchased.
pay-as-you-goPrice	All	Retail price for the resource.
PartnerEarnedCreditApplied	MPA	Indicates whether the partner earned credit has been applied.
PartnerEarnedCreditRate	MPA	Rate of discount applied if there's a partner earned credit (PEC), based on partner admin link access.
PartnerName	MPA	Name of the partner Azure Active Directory tenant.
PartnerTenantId	MPA	Identifier for the partner's Azure Active Directory tenant.
PartNumber <sup>1</sup>	EA, pay-as-you-go	Identifier used to get specific meter pricing.
PlanName	EA, pay-as-you-go	Marketplace plan name.
PreviousInvoiceId	MCA	Reference to an original invoice if the line item is a refund.
PricingCurrency	MCA	Currency used when rating based on negotiated prices.
PricingModel	All	Identifier that indicates how the meter is priced. (Values: <code>On Demand</code> , <code>Reservation</code> , and <code>Spot</code> )
Product	All	Name of the product.
ProductId <sup>1</sup>	MCA	Unique identifier for the product.
ProductOrderId	All	Unique identifier for the product order.
ProductOrderName	All	Unique name for the product order.
Provider	All	Identifier for product category or Line of Business. For example, Azure, Microsoft 365, and AWS.
PublisherId	MCA	The ID of the publisher. It's only available after the invoice is generated.
PublisherName	All	Publisher for Marketplace services.

Term	Account type	Description
PublisherType	All	Supported values: <b>Microsoft</b> , <b>Azure</b> , <b>AWS</b> , <b>Marketplace</b> . Values are <b>Microsoft</b> for MCA accounts and <b>Azure</b> for EA and pay-as-you-go accounts.
Quantity	All	The number of units purchased or consumed.
ResellerName	MPA	The name of the reseller associated with the subscription.
ResellerMpnId	MPA	ID for the reseller associated with the subscription.
ReservationId	EA, MCA	Unique identifier for the purchased reservation instance.
ReservationName	EA, MCA	Name of the purchased reservation instance.
ResourceGroup	All	Name of the <a href="#">resource group</a> the resource is in. Not all charges come from resources deployed to resource groups. Charges that don't have a resource group will be shown as null or empty, <b>Others</b> , or <b>Not applicable</b> .
ResourceId <sup>1</sup>	All	Unique identifier of the <a href="#">Azure Resource Manager</a> resource.
ResourceLocation <sup>1</sup>	All	Datacenter location where the resource is running. See <a href="#">Location</a> .
ResourceName	EA, pay-as-you-go	Name of the resource. Not all charges come from deployed resources. Charges that don't have a resource type will be shown as null/empty, <b>Others</b> , or <b>Not applicable</b> .
ResourceType	MCA	Type of resource instance. Not all charges come from deployed resources. Charges that don't have a resource type will be shown as null/empty, <b>Others</b> , or <b>Not applicable</b> .
RoundingAdjustment	EA, MCA	Rounding adjustment represents the quantization that occurs during cost calculation. When the calculated costs are converted to the invoiced total, small rounding errors can occur. The rounding errors are represented as <a href="#">rounding adjustment</a> to ensure that the costs shown in Cost Management align to the invoice.
ServiceFamily	MCA	Service family that the service belongs to.
ServiceInfo <sup>1</sup>	All	Service-specific metadata.

<b>Term</b>	<b>Account type</b>	<b>Description</b>
ServiceInfo2	All	Legacy field with optional service-specific metadata.
ServicePeriodEndDate	MCA	The end date of the rating period that defined and locked pricing for the consumed or purchased service.
ServicePeriodStartDate	MCA	The start date of the rating period that defined and locked pricing for the consumed or purchased service.
SubscriptionId <sup>1</sup>	All	Unique identifier for the Azure subscription.
SubscriptionName	All	Name of the Azure subscription.
Tags <sup>1</sup>	All	Tags assigned to the resource. Doesn't include resource group tags. Can be used to group or distribute costs for internal chargeback. For more information, see <a href="#">Organize your Azure resources with tags</a> .
Term	All	Displays the term for the validity of the offer. For example: In case of reserved instances, it displays 12 months as the Term. For one-time purchases or recurring purchases, Term is one month (SaaS, Marketplace Support). Not applicable for Azure consumption.
UnitOfMeasure	All	The unit of measure for billing for the service. For example, compute services are billed per hour.
UnitPrice	EA, pay-as-you-go	The price per unit for the charge.

<sup>1</sup> Fields used to build a unique ID for a single cost record. Every record in your cost details file should be considered unique.

The cost details file itself doesn't uniquely identify individual records with an ID. Instead, you can use fields in the file flagged with <sup>1</sup> to create a unique ID yourself.

Some fields might differ in casing and spacing between account types. Older versions of pay-as-you-go cost details files have separate sections for the statement and daily cost.

## List of terms from older APIs

The following table maps terms used in older APIs to the new terms. Refer to the above table for those descriptions.

<b>Old term</b>	<b>New term</b>
ConsumedQuantity	Quantity
IncludedQuantity	N/A
InstanceId	ResourceId
Rate	EffectivePrice
Unit	UnitOfMeasure
UsageDate	Date
UsageEnd	Date
UsageStart	Date

## Next steps

- Get an overview of how to [ingest cost data](#).
- Learn more about [Choose a cost details solution](#).
- [Create and manage exported data](#) in the Azure portal with Exports.
- [Automate Export creation](#) and ingestion at scale using the API.
- Learn how to [Get small cost datasets on demand](#).

# Retrieve large cost datasets recurrently with exports

Article • 07/17/2022

This article helps you regularly export large amounts of data with exports from Cost Management. Exporting is the recommended way to retrieve unaggregated cost data. Especially when usage files are too large to reliably call and download using the [Cost Details API](#). Exported data is placed in the Azure Storage account that you choose. From there, you can load it into your own systems and analyze it as needed. To configure exports in the Azure portal, see [Export data](#).

If you want to automate exports at various scopes, the sample API request in the next section is a good starting point. You can use the Exports API to create automatic exports as a part of your general environment configuration. Automatic exports help ensure that you have the data that you need. You can use in your own organization's systems as you expand your Azure use.

## Common export configurations

Before you create your first export, consider your scenario and the configuration options need to enable it. Consider the following export options:

- **Recurrence** - Determines how frequently the export job runs and when a file is put in your Azure Storage account. Choose between Daily, Weekly, and Monthly. Try to configure your recurrence to match the data import jobs used by your organization's internal system.
- **Recurrence Period** - Determines how long the Export remains valid. Files are only exported during the recurrence period.
- **Time Frame** - Determines the amount of data that's generated by the export on a given run. Common options are MonthToDate and WeekToDate.
- **StartDate** - Configures when you want the export schedule to begin. An export is created on the StartDate and then later based on your Recurrence.
- **Type** - There are three export types:
  - ActualCost - Shows the total usage and costs for the period specified, as they're accrued and shows on your bill.
  - AmortizedCost - Shows the total usage and costs for the period specified, with amortization applied to the reservation purchase costs that are applicable.
  - Usage - All exports created before July 20 2020 are of type Usage. Update all your scheduled exports as either ActualCost or AmortizedCost.

- **Columns** – Defines the data fields you want included in your export file. They correspond with the fields available in the [Cost Details API](#).
- **Partitioning** - Set the option to true if you have a large dataset and would like it to be broken up into multiple files. This makes data ingestion much faster and easier. For more information about partitioning, see [File partitioning for large datasets](#).

## Create a daily month-to-date export for a subscription

Request URL: `PUT`

```
https://management.azure.com/{scope}/providers/Microsoft.CostManagement/exports/{exportName}?api-version=2020-06-01
```

JSON

```
{
 "properties": {
 "schedule": {
 "status": "Active",
 "recurrence": "Daily",
 "recurrencePeriod": {
 "from": "2020-06-01T00:00:00Z",
 "to": "2020-10-31T00:00:00Z"
 }
 },
 "format": "Csv",
 "deliveryInfo": {
 "destination": {
 "resourceId": "/subscriptions/00000000-0000-0000-0000-000000000000/resourceGroups/MYDEVTESTRG/providers/Microsoft.Storage/storageAccounts/{yourStorageAccount} ",
 "container": "{yourContainer}",
 "rootFolderPath": "{yourDirectory}"
 }
 },
 "definition": {
 "type": "ActualCost",
 "timeframe": "MonthToDate",
 "dataSet": {
 "granularity": "Daily",
 "configuration": {
 "columns": [
 "Date",
 "MeterId",
 "ResourceId",
 "ResourceLocation",
 "Quantity"
]
 }
 }
 }
}
```

```
 }
}
}
```

## Copy large Azure storage blobs

You can use Cost Management to schedule exports of your Azure usage details into your Azure Storage accounts as blobs. The resulting blob sizes could be over gigabytes in size. The Cost Management team worked with the Azure Storage team to test copying large Azure storage blobs. The results are documented in the following sections. You can expect to have similar results as you copy storage blobs from one Azure region to another.

To test its performance, the team transferred blobs from storage accounts in the US West region to the same and other regions. The team measured speeds that ranged from 2 GB per second in the same region to 150 MB per second to storage accounts in the South East Asia region.

## Test configuration

To measure blob transfer speeds, the team created a simple .NET console application referencing the latest version (v2.0.1) of the Azure Data Movement Library (DLM) via NuGet. DLM is an SDK provided by the Azure Storage team that enables programmatic access to their transfer services. Then they created Standard V2 storage accounts in multiple regions and use the West US as the source region. They populated the storage accounts there with containers, where each held ten 2-GB block blobs. They copied the containers to other storage accounts using DLM's *TransferManager.CopyDirectoryAsync()* method with the *CopyMethod.ServiceSideSyncCopy* option. Tests were conducted on a computer running Windows 10 with 12 cores and 1-GbE network.

Application settings used:

- *TransferManager.Configurations.ParallelOperations = Environment.ProcessorCount \* 32*. The team found the setting to have the most effect on overall throughput. A value of 32 times the number of cores provided the best throughput for the test client.
- *ServicePointManager.DefaultConnectionLimit = int.MaxValue*. Setting it to a maximum value effectively passes full control of transfer parallelism to the *ParallelOperations* setting above.
- *TransferManager.Configurations.BlockSize = 4,194,304*. It had some effect on transfer rates with 4 MB, proving to be best for testing.

For more information and sample code, see links in the [Next steps](#) section.

## Test results

Test number	To region	Blobs	Time (secs)	MB/s	Comments
1	WestUS	2 GB x 10	10	2,000	
2	WestUS2	2 GB x 10	33	600	
3	EastUS	2 GB x 10	67	300	
4	EastUS	2 GB x 10 x 4	99	200	4 parallel transfers using 8 storage accounts: 4 West to 4 East average per transfer
6	EastUS	2 GB x 10 x 4	92	870	4 parallel transfers from 1 storage account to another
5	EastUS	2G x 10 x 8	148	135	8 parallel transfers using 8 storage accounts: 4 West to 4x2 East average per transfer
7	SE Asia	2 GB x 10	133	150	
8	SE Asia	2 GB x 10 x 4	444	180	4 parallel transfers from 1 storage account to another

## Sync transfer characteristics

Here are some of the characteristics of the service-side sync transfer used with DML that are relevant to its use:

- DML can transfer a single blob or a directory. For directory transfer, you can use a search pattern to match on blob prefix.
- Block blob transfers happen in parallel. All complete towards the end of the transfer process. Individual blob blocks are transferred in parallel.
- The transfer is executed asynchronously on the client. The transfer status is available periodically via a callback to a method that can be defined in a *TransferContext* object.
- The transfer creates checkpoints during its progress and exposes a *TransferCheckpoint* object. The object represents the latest checkpoint via the *TransferContext* object. If the *TransferCheckpoint* is saved before a transfer is

- cancelled/aborted, the transfer can be resumed from the checkpoint for up to seven days. The transfer can be resumed from any checkpoint, not just the latest.
- If the transfer client process is killed and restarted without implementing the checkpoint feature.
    - Before any blob transfers have been completed, the transfer restarts.
    - After some of the blobs have been completed, the transfer restarts for only the incompletely transferred blobs.
  - Pausing the client execution pauses the transfers.
  - The blob transfer feature abstracts the client from transient failures. For instance, storage account throttling won't normally cause a transfer to fail but will slow the transfer.
  - Service-side transfers have low client resource usage for CPU and memory, some network bandwidth, and connections.

## Async transfer characteristics

You can invoke the `TransferManager.CopyDirectoryAsync()` method with the `CopyMethod.ServiceSideAsyncCopy` option. It operates similar to the sync transfer mechanism from the client perspective but with the following differences in operation:

- Transfer rates are much slower than the equivalent sync transfer (typically 10 MB/s or less).
- The transfer continues even if the client process terminates.
- Although checkpoints are supported, resuming a transfer using a `TransferCheckpoint` won't resume at the checkpoint time but at the current state of the transfer.

## Test summary

Azure blob storage supports high global transfer rates with its service-side sync transfer feature. Using the feature in .NET applications is straightforward using the Data Movement Library. It's possible for Cost Management exports to reliably copy hundreds of gigabytes of data to a storage account anywhere in less than an hour.

## Next steps

- See the [Microsoft Azure Storage Data Movement Library](#) source.
- [Transfer data with the Data Movement library](#).
- See the [AzureDmlBackup sample application](#) source sample.
- Read [High-Throughput with Azure Blob Storage](#).

# Get small cost datasets on demand

Article • 05/10/2023

Use the [Cost Details API](#) to get raw, unaggregated cost data that corresponds to your Azure bill. The API is useful when your organization needs a programmatic data retrieval solution. Consider using the API if want to analyze smaller cost data sets of 2 GB (2 million rows) or less. However, you should use Exports for ongoing data ingestion workloads and for the download of larger datasets.

If you want to get large amounts of exported data regularly, see [Retrieve large cost datasets recurrently with exports](#).

To learn more about the data in cost details (formerly referred to as *usage details*), see [Ingest cost details data](#).

The [Cost Details report](#) is only available for customers with an Enterprise Agreement or Microsoft Customer Agreement. If you're an MSDN, Pay-As-You-Go or Visual Studio customer, see [Get cost details for a pay-as-you-go subscription](#).

## Permissions

To use the Cost Details API, you need read only permissions for supported features and scopes.

### Note

The [Cost Details API](#) doesn't support management groups for either EA or MCA customers.

For more information, see:

- [Azure RBAC scopes - role permissions for feature behavior](#)
- [Enterprise Agreement scopes - role permissions for feature behavior](#)
- [Microsoft Customer Agreement scopes - role permissions for feature behavior](#)

## Cost Details API best practices

Microsoft recommends the following best practices as you use the Cost Details API.

### Request schedule

If you want to get the latest cost data, we recommend you query at most once per day. Reports are refreshed every four hours. If you call more frequently, you'll receive identical data. Once you download your cost data for historical invoices, the charges won't change unless you're explicitly notified. We recommend caching your cost data in a queryable store on your side to prevent repeated calls for identical data.

## Chunk your requests

Chunk your calls into small date ranges to get more manageable files that you can download over the network. For example, we recommend chunking by day or by week if you have a large Azure cost file month-to-month. If you have scopes with a large amount of cost data (for example a Billing Account), consider placing multiple calls to child scopes so you get more manageable files that you can download. For more information about Cost Management scopes, see [Understand and work with scopes](#). After you download the data, use Excel to analyze data further with filters and pivot tables.

If your dataset is more than 2 GB (or roughly 2 million rows) month-to-month, consider using [Exports](#) as a more scalable solution.

## Latency and rate limits

On demand calls to the API are rate limited. The time it takes to generate your cost details file is directly correlated with the amount of data in the file. To understand the expected amount of time before your file becomes available for download, you can use the `retry-after` header in the API response.

## Supported dataset time ranges

The Cost Details API supports a maximum data set time range of one month per report. Historical data can be retrieved for up to 13 months back from the current date. If you're looking to seed a 13 month historical dataset, we recommend placing 13 calls for one month datasets going back 13 months.

## Example Cost Details API requests

The following example requests are used by Microsoft customers to address common scenarios. The data that's returned by the request corresponds to the date when the cost was received by the billing system. It might include costs from multiple invoices. It's an asynchronous API. As such, you place an initial call to request your report and receive

a polling link in the response header. From there, you can poll the link provided until the report is available for you.

Use the `retry-after` header in the API response to dictate when to poll the API next. The header provides an estimated minimum time that your report will take to generate.

To learn more about the API contract, see [Cost Details API](#).

## Actual cost versus amortized cost

To control whether you would like to see an actual cost or amortized cost report, change the value used for the metric field in the initial request body. The available metric values are `ActualCost` or `AmortizedCost`.

Amortized cost breaks down your reservation purchases into daily chunks and spreads them over the duration of the reservation term. For example, instead of seeing a \$365 purchase on January 1, you'll see a \$1.00 purchase every day from January 1 to December 31. In addition to basic amortization, the costs are also reallocated and associated by using the specific resources that used the reservation. For example, if the \$1.00 daily charge was split between two virtual machines, you'd see two \$0.50 charges for the day. If part of the reservation isn't utilized for the day, you'd see one \$0.50 charge associated with the applicable virtual machine and another \$0.50 charge with a charge type of `UnusedReservation`. Unused reservation costs are seen only when viewing amortized cost.

Because of the change in how costs are represented, it's important to note that actual cost and amortized cost views will show different total numbers. In general, the total cost of months over time for a reservation purchase will decrease when viewing amortized costs. The months following a reservation purchase will increase. Amortization is available only for reservation purchases and doesn't currently apply to Azure Marketplace purchases.

## Initial request to create report

HTTP

POST

`https://management.azure.com/{scope}/providers/Microsoft.CostManagement/generateCostDetailsReport?api-version=2022-05-01`

Request body:

An example request for an ActualCost dataset for a specified date range is provided below.

```
JSON

{
 "metric": "ActualCost",
 "timePeriod": {
 "start": "2020-03-01",
 "end": "2020-03-15"
 }
}
```

Available {scope} options to build the proper URI are documented at [Identify the resource ID for a scope](#).

The available fields you can provide in the report request body are summarized below.

- **metric** - The type of report requested. It can be either ActualCost or AmortizedCost. Not required. If the field isn't specified, the API will default to an ActualCost report.
- **timePeriod** - The requested date range for your data. Not required. This parameter can't be used alongside either the invoiceId or billingPeriod parameters. If a timePeriod, invoiceId or billingPeriod parameter isn't provided in the request body the API will return the current month's cost.
- **invoiceId** - The requested invoice for your data. This parameter can only be used by Microsoft Customer Agreement customers. Additionally, it can only be used at the Billing Profile or Customer scope. This parameter can't be used alongside either the billingPeriod or timePeriod parameters. If a timePeriod, invoiceId or billingPeriod parameter isn't provided in the request body the API will return the current month's cost.
- **billingPeriod** - The requested billing period for your data. This parameter can be used only by Enterprise Agreement customers. Use the YearMonth format. For example, 202008. This parameter can't be used alongside either the invoiceId or timePeriod parameters. If a timePeriod, invoiceId or billingPeriod parameter isn't provided in the request body the API will return the current month's cost.

#### API response:

**Response Status: 202 – Accepted** : Indicates that the request will be processed. Use the **Location** header to check the status.

#### Response headers:

Name	Type	Format	Description
Location	String		The URL to check the result of the asynchronous operation.
Retry-After	Integer	Int32	The expected time for your report to be generated. Wait for this duration before polling again.

## Report polling and download

Once you've requested to create a Cost Details report, poll for the report using the endpoint provided in the `location` header of the API response. An example polling request is below.

Report polling request:

```
HTTP
GET
https://management.azure.com/{scope}/providers/Microsoft.CostManagement/cost
DetailsOperationStatus/{operationId}?api-version=2022-05-01
```

`Response Status 200 – Succeeded`: Indicates that the request has succeeded.

```
JSON
{
 "id": "subscriptions/00000000-0000-0000-0000-
000000000000/providers/Microsoft.CostManagement/operationResults/00000000-
0000-0000-0000-000000000000",
 "name": "00000000-0000-0000-0000-000000000000",
 "status": "Completed",
 "manifest": {
 "manifestVersion": "2022-05-01",
 "dataFormat": "Csv",
 "blobCount": 1,
 "byteCount": 160769,
 "compressData": false,
 "requestContext": {
 "requestScope": "subscriptions/00000000-0000-0000-000000000000",
 "requestBody": {
 "metric": "ActualCost",
 "timePeriod": {
 "start": "2020-03-01",
 "end": "2020-03-15"
 }
 }
 },
 "blobs": [
 {
 "blobType": "ReportManifest"
 }
]
}
```

```
 "blobLink": "{downloadLink}",
 "byteCount": 32741
 }
]
},
"validTill": "2022-05-10T08:08:46.197325Z"
}
```

A summary of the key fields in the API response is below:

- **manifestVersion** - The version of the manifest contract that is used in the response. At this time, the manifest version will remain the same for a given API version.
- **dataFormat** - CSV is the only supported file format provided by the API at this time.
- **blobCount** - Represents the number of individual data blobs present in the report dataset. It's important to note that this API may provide a partitioned dataset of more than one file in the response. Design your data pipelines to be able to handle partitioned files accordingly. Partitioning will allow you to be able to ingest larger datasets more quickly moving forward.
- **byteCount** - The total byte count of the report dataset across all partitions.
- **compressData** - Compression is always set to false for the first release. The API will support compression in the future, however.
- **requestContext** - The initial configuration requested for the report.
- **blobs** - A list of n blob files that together comprise the full report.
  - **blobLink** - The download URL of an individual blob partition.
  - **byteCount** - The byte count of the individual blob partition.
- **validTill** - The date at which the report will no longer be accessible.

## Next steps

- Read the [Ingest cost details data](#) article.
- Learn more about [Choose a cost details solution](#).
- [Understand cost details fields](#).
- [Create and manage exported data](#) in the Azure portal with exports.
- [Automate Export creation](#) and ingestion at scale using the API.

# Get cost details for a pay-as-you-go subscription

Article • 07/18/2022

If you have an MSDN, Microsoft Online Service Program (MOSP) pay-as-you-go, or Visual Studio Azure subscription, we recommend that you use [Exports](#) or the [Exports API](#) to get cost details data (formerly known as usage details). The [Cost Details API](#) report isn't supported for your subscription type yet.

If you need to download small datasets and you don't want to use Azure Storage, you can also use the Consumption Usage Details API. Instructions about how to use the API are below.

## ⓘ Note

The API is deprecated for all customers except those with MSDN, pay-as-you-go and Visual Studio subscriptions. If you're an EA or MCA customer don't use this API.

The date that the API will be turned off is still being determined. The [Cost Details API](#) will be updated to support MSDN, pay-as-you-go and Visual studio subscriptions prior to the deprecation of the Consumption Usage Details API.

## Example Consumption Usage Details API requests

The following example requests are used by Microsoft customers to address common scenarios.

### Get usage details for a scope during a specific date range

The data that's returned by the request corresponds to the date when the data was received by the billing system. It might include costs from multiple invoices. The call to use varies by your subscription type.

For pay-as-you-go subscriptions, use the following call.

HTTP

```
GET
```

```
https://management.azure.com/{scope}/providers/Microsoft.Consumption/usageDe-
tails?$filter=properties%2FusageStart%20ge%20'2020-02-
01'%20and%20properties%2FusageEnd%20le%20'2020-02-29'&$top=1000&api-
version=2019-10-01
```

## Get amortized cost details

If you need actual costs to show purchases as they're accrued, change the `metric` to `ActualCost` in the following request. To use amortized and actual costs, you must use version `2019-04-01-preview` or later.

```
HTTP
```

```
GET
```

```
https://management.azure.com/{scope}/providers/Microsoft.Consumption/usageDe-
tails?metric=AmortizedCost&$filter=properties/usageStart+ge+'2019-04-
01'+AND+properties/usageEnd+le+'2019-04-30'&api-version=2019-04-01-preview
```

## Next steps

- Read the [Ingest cost details data](#) article.
- Learn how to [Get small cost datasets on demand](#).
- [Understand cost details fields](#).
- [Create and manage exported data](#) in the Azure portal with exports.
- [Automate Export creation](#) and ingestion at scale using the API.

# Get usage data with the Azure CLI

Article • 04/13/2023

This article explains how you get cost and usage data with the Azure CLI. If you want to get usage data using the Azure portal, see [View and download your Azure usage and charges](#).

## Set up the Azure CLI

Start by preparing your environment for the Azure CLI.

- Use the Bash environment in [Azure Cloud Shell](#). For more information, see [Quickstart for Bash in Azure Cloud Shell](#).  
A blue rectangular button with a white 'A' icon and the text 'Launch Cloud Shell' next to it, with a small blue arrow pointing to the right at the end of the text.  
A blue rectangular button with a white 'A' icon and the text 'Launch Cloud Shell' next to it, with a small blue arrow pointing to the right at the end of the text.
- If you prefer to run CLI reference commands locally, [install](#) the Azure CLI. If you're running on Windows or macOS, consider running Azure CLI in a Docker container. For more information, see [How to run the Azure CLI in a Docker container](#).
  - If you're using a local installation, sign in to the Azure CLI by using the [az login](#) command. To finish the authentication process, follow the steps displayed in your terminal. For other sign-in options, see [Sign in with the Azure CLI](#).
  - When you're prompted, install the Azure CLI extension on first use. For more information about extensions, see [Use extensions with the Azure CLI](#).
  - Run [az version](#) to find the version and dependent libraries that are installed. To upgrade to the latest version, run [az upgrade](#).

## Configure an export job to export cost data to Azure storage

After you sign in, use the [export](#) commands to export usage data to an Azure storage account. You can download the data from there.

1. Create a resource group or use an existing resource group. To create a resource group, run the [group create](#) command:

Azure CLI

```
az group create --name TreyNetwork --location "East US"
```

2. Create a storage account to receive the exports or use an existing storage account.

To create an account, use the [storage account create](#) command:

Azure CLI

```
az storage account create --resource-group TreyNetwork --name cmdemo
```

3. Run the [export create](#) command to create the export:

Azure CLI

```
az costmanagement export create --name DemoExport --type Usage \--scope "subscriptions/00000000-0000-0000-0000-000000000000" --storage-account-id cmdemo \--storage-container democontainer --timeframe MonthToDate --storage-directory demodirectory
```

## Next steps

- Read the [Ingest usage details data](#) article.
- Learn how to [Get small cost datasets on demand](#).
- [Understand usage details fields](#).
- Create and manage [exported data](#) in the Azure portal with exports.
- Automate [Export creation](#) and ingestion at scale using the API.

# Automate budget creation

Article • 03/26/2023

You can automate budget creation using the [Budgets API](#). You can also create a budget with a [budget template](#). Templates are an easy way for you to standardize Azure deployments while ensuring cost control is properly configured and enforced.

## Common Budgets API configurations

There are many ways to configure a budget in your Azure environment. Consider your scenario first and then identify the configuration options that enable it. Review the following options:

- **Time Grain** - Represents the recurring period your budget uses to accrue and evaluate costs. The most common options are Monthly, Quarterly, and Annual.
- **Time Period** - Represents how long your budget is valid. The budget actively monitors and alerts you only while it remains valid.
- **Notifications**
  - Contact Emails – The email addresses receive alerts when a budget accrues costs and exceeds defined thresholds.
  - Contact Roles - All users who have a matching Azure role on the given scope receive email alerts with this option. For example, Subscription Owners could receive an alert for a budget created at the subscription scope.
  - Contact Groups - The budget calls the configured action groups when an alert threshold is exceeded.
- **Cost dimension filters** - The same filtering you can do in Cost Analysis or the Query API can also be done on your budget. Use this filter to reduce the range of costs that you're monitoring with the budget.

After you've identified the budget creation options that meet your needs, create the budget using the API. The example below helps get you started with a common budget configuration.

## Create a budget filtered to multiple resources and tags

Request URL: `PUT`

```
https://management.azure.com/subscriptions/{SubscriptionId}/providers/Microsoft.Con
sumption/budgets/{BudgetName}/?api-version=2019-10-01
```

JSON

```
{
 "eTag": "\"1d34d016a593709\"",
 "properties": {
 "category": "Cost",
 "amount": 100.65,
 "timeGrain": "Monthly",
 "timePeriod": {
 "startDate": "2017-10-01T00:00:00Z",
 "endDate": "2018-10-31T00:00:00Z"
 },
 "filter": {
 "and": [
 {
 "dimensions": {
 "name": "ResourceId",
 "operator": "In",
 "values": [
 "/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/virtualMachines/{meterName}",
 "/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}/providers/Microsoft.Compute/virtualMachines/{meterName}"
]
 }
 },
 {
 "tags": {
 "name": "category",
 "operator": "In",
 "values": [
 "Dev",
 "Prod"
]
 }
 },
 {
 "tags": {
 "name": "department",
 "operator": "In",
 "values": [
 "engineering",
 "sales"
]
 }
 }
]
 },
 "notifications": {
 "Actual_GreaterThan_80_Percent": {
 "enabled": true,
 "operator": "GreaterThan",
 "threshold": 80,
 "contactEmails": [
 "mailto:user1@example.com",
 "mailto:user2@example.com"
]
 }
 }
 }
}
```

```

 "user1@contoso.com",
 "user2@contoso.com"
],
 "contactRoles": [
 "Contributor",
 "Reader"
],
 "contactGroups": [
 "/subscriptions/{subscriptionID}/resourceGroups/{resourceGroupName}/providers/microsoft.insights/actionGroups/{actionGroupName}"
],
 "thresholdType": "Actual"
}
}
}
}

```

## Supported locales for budget alert emails

With budgets, you're alerted when costs cross a set threshold. You can set up to five email recipients per budget. Recipients receive the email alerts within 24 hours of crossing the budget threshold. However, your recipient might need to receive an email in a different language. You can use the following language culture codes with the Budgets API. Set the culture code with the `locale` parameter similar to the following example.

JSON

```
{
 "eTag": "\"1d681a8fc67f77a\"",
 "properties": {
 "timePeriod": {
 "startDate": "2020-07-24T00:00:00Z",
 "endDate": "2022-07-23T00:00:00Z"
 },
 "timeGrain": "BillingMonth",
 "amount": 1,
 "currentSpend": {
 "amount": 0,
 "unit": "USD"
 },
 "category": "Cost",
 "notifications": {
 "actual_GreaterThan_10_Percent": {
 "enabled": true,
 "operator": "GreaterThan",
 "threshold": 20,
 "locale": "en-us",
 "contactEmails": [

```

```

 "user@contoso.com"
],
 "contactRoles": [],
 "contactGroups": [],
 "thresholdType": "Actual"
}
}
}
}
```

Languages supported by a culture code:

Culture code	Language
en-us	English (United States)
ja-jp	Japanese (Japan)
zh-cn	Chinese (Simplified, China)
de-de	German (Germany)
es-es	Spanish (Spain, International)
fr-fr	French (France)
it-it	Italian (Italy)
ko-kr	Korean (Korea)
pt-br	Portuguese (Brazil)
ru-ru	Russian (Russia)
zh-tw	Chinese (Traditional, Taiwan)
cs-cz	Czech (Czech Republic)
pl-pl	Polish (Poland)
tr-tr	Turkish (Türkiye)
da-dk	Danish (Denmark)
en-gb	English (United Kingdom)
hu-hu	Hungarian (Hungary)
nb-no	Norwegian Bokmal (Norway)
nl-nl	Dutch (Netherlands)
pt-pt	Portuguese (Portugal)

Culture code	Language
sv-se	Swedish (Sweden)

# Configure cost-based orchestration for budget alerts

You can configure budgets to start automated actions using Azure Action Groups. To learn more about automating actions using budgets, see [Automation with budgets](#).

## Next steps

- Learn more about Cost Management + Billing automation at [Cost Management automation overview](#).
- Assign permissions to [Cost Management APIs](#).

# Manage costs with budgets

Article • 12/07/2022

Cost control is a critical component to maximizing the value of your investment in the cloud. There are several scenarios where cost visibility, reporting, and cost-based orchestration are critical to continued business operations. [Cost Management APIs](#) provide a set of APIs to support each of these scenarios. The APIs provide usage details, allowing you to view granular instance level costs.

Budgets are commonly used as part of cost control. Budgets can be scoped in Azure. For instance, you could narrow your budget view based on subscription, resource groups, or a collection of resources. In addition to using the budgets API to notify you via email when a budget threshold is reached, you can use [Azure Monitor action groups](#) to trigger an orchestrated set of actions resulting from a budget event.

A common budgets scenario for a customer running a non-critical workload could occur when they want to manage against a budget and also get to a predictable cost when looking at the monthly invoice. This scenario requires some cost-based orchestration of resources that are part of the Azure environment. In this scenario, a monthly budget of \$1000 for the subscription is set. Also, notification thresholds are set to trigger a few orchestrations. This scenario starts with an 80% cost threshold, which will stop all VMs in the resource group **Optional**. Then, at the 100% cost threshold, all VM instances will be stopped.

To configure this scenario, you'll complete the following actions by using the steps provided in each section of this tutorial.

These actions included in this tutorial allow you to:

- Create an Azure Automation Runbook to stop VMs by using webhooks.
- Create an Azure Logic App to be triggered based on the budget threshold value and call the runbook with the right parameters.
- Create an Azure Monitor Action Group that will be configured to trigger the Azure Logic App when the budget threshold is met.
- Create the budget with the wanted thresholds and wire it to the action group.

## Create an Azure Automation Runbook

[Azure Automation](#) is a service that enables you to script most of your resource management tasks and run those tasks as either scheduled or on-demand. As part of this scenario, you'll create an [Azure Automation runbook](#) that will be used to stop VMs.

You'll use the [Stop Azure V2 VMs](#) graphical runbook from the [gallery](#) to build this scenario. By importing this runbook into your Azure account and publishing it, you can stop VMs when a budget threshold is reached.

## Create an Azure Automation account

1. Sign in to the [Azure portal](#) with your Azure account credentials.
2. Select the **Create a resource** button found on the upper left corner of Azure.
3. Select **Management Tools > Automation**.

 **Note**

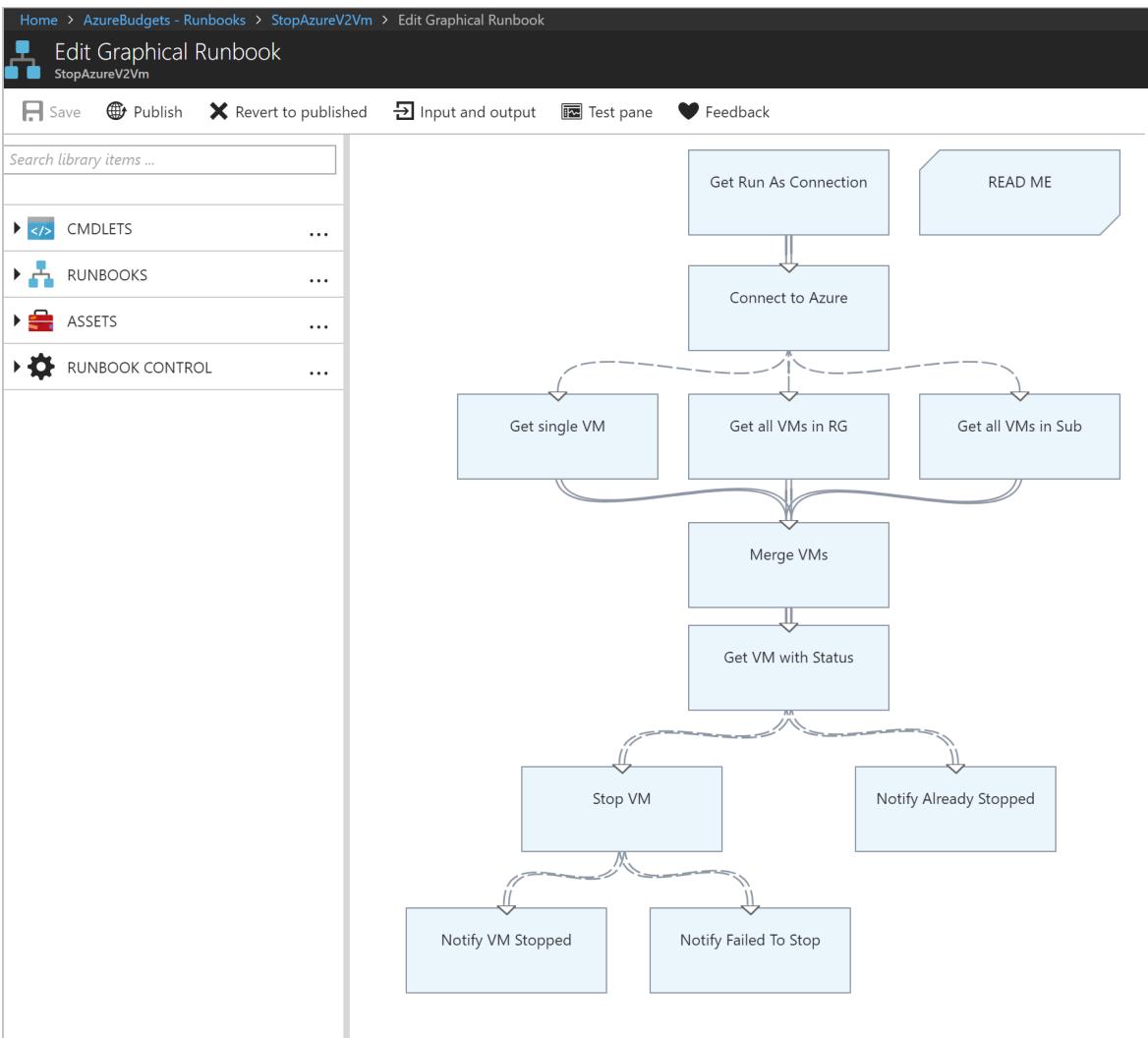
If you don't have an Azure account, you can create a [free account](#).

4. Enter your account information. For **Create Azure Run As account**, choose **Yes** to automatically enable the settings needed to simplify authentication to Azure.
5. When complete, select **Create**, to start the Automation account deployment.

## Import the Stop Azure V2 VMs runbook

Using an [Azure Automation runbook](#), import the [Stop Azure V2 VMs](#) graphical runbook from the gallery.

1. Sign in to the [Azure portal](#) with your Azure account credentials.
2. Open your Automation account by selecting **All services > Automation Accounts**. Then, select your Automation Account.
3. Select **Runbooks gallery** from the **Process Automation** section.
4. Set the **Gallery Source** to **Script Center** and select **OK**.
5. Locate and select the [Stop Azure V2 VMs](#) gallery item within the Azure portal.
6. Select **Import** to display the **Import** area and select **OK**. The runbook overview area will be displayed.
7. Once the runbook has completed the import process, select **Edit** to display the graphical runbook editor and publishing option.



8. Select **Publish** to publish the runbook and then select **Yes** when prompted. When you publish a runbook, you override any existing published version with the draft version. In this case, you've no published version because you've created the runbook. For more information about publishing a runbook, see [Create a graphical runbook](#).

## Create webhooks for the runbook

Using the [Stop Azure V2 VMs](#) graphical runbook, you create two Webhooks to start the runbook in Azure Automation through a single HTTP request. The first webhook invokes the runbook at an 80% budget threshold with the resource group name as a parameter, allowing the optional VMs to be stopped. Then, the second webhook invokes the runbook with no parameters (at 100%), which stops all remaining VM instances.

1. From the **Runbooks** page in the [Azure portal](#), select the **StopAzureV2Vm** runbook that displays the runbook's overview area.
2. Select **Webhook** at the top of the page to open the **Add Webhook** area.
3. Select **Create new webhook** to open the **Create a new webhook** area.

- Set the Name of the Webhook to **Optional**. The Enabled property must be **Yes**. You don't need to change the Expires value. For more information about Webhook properties, see [Webhook properties](#).
- Next to the URL value, select the copy icon to copy the URL of the webhook.

 **Important**

Save the URL of the webhook named **Optional** in a safe place. You'll use the URL later in this tutorial. For security reasons, once you create the webhook, you cannot view or retrieve the URL again.

- Select **OK** to create the new webhook.
- Select **Configure parameters and run settings** to view parameter values for the runbook.

 **Note**

If the runbook has mandatory parameters, then you are not able to create the webhook unless values are provided.

- Select **OK** to accept the webhook parameter values.
- Select **Create** to create the webhook.
- Next, follow the steps above to create a second webhook named **Complete**.

 **Important**

Be sure to save both webhook URLs to use later in this tutorial. For security reasons, once you create the webhook, you cannot view or retrieve the URL again.

You should now have two configured webhooks that are each available using the URLs that you saved.

Name	Expiration	Last Triggered	Status
Complete	7/18/2019, 4:06 PM		✓ Enabled
Optional	7/11/2019, 11:03 AM		✓ Enabled

You're now done with the Azure Automation setup. You can test the webhooks with a simple Postman test to validate that the webhook works. Next, you must create the Logic App for orchestration.

## Create an Azure Logic App for orchestration

Logic Apps helps you build, schedule, and automate processes as workflows so you can integrate apps, data, systems, and services across enterprises or organizations. In this scenario, the [Logic App](#) you create will do a little more than just call the automation webhook you created.

Budgets can be set up to trigger a notification when a specified threshold is met. You can provide multiple thresholds to be notified at and the Logic App will demonstrate the ability for you to perform different actions based on the threshold met. In this example, you'll set up a scenario where you get a couple of notifications, the first notification is for when 80% of the budget has been reached and the second notification is when 100% of the budget has been reached. The logic app will be used to shut down all VMs in the resource group. First, the **Optional** threshold will be reached at 80%, then the second threshold will be reached where all VMs in the subscription will be shut down.

Logic apps allow you to provide a sample schema for the HTTP trigger, but require you to set the **Content-Type** header. Because the action group doesn't have custom headers for the webhook, you must parse out the payload in a separate step. You'll use the **Parse** action and provide it with a sample payload.

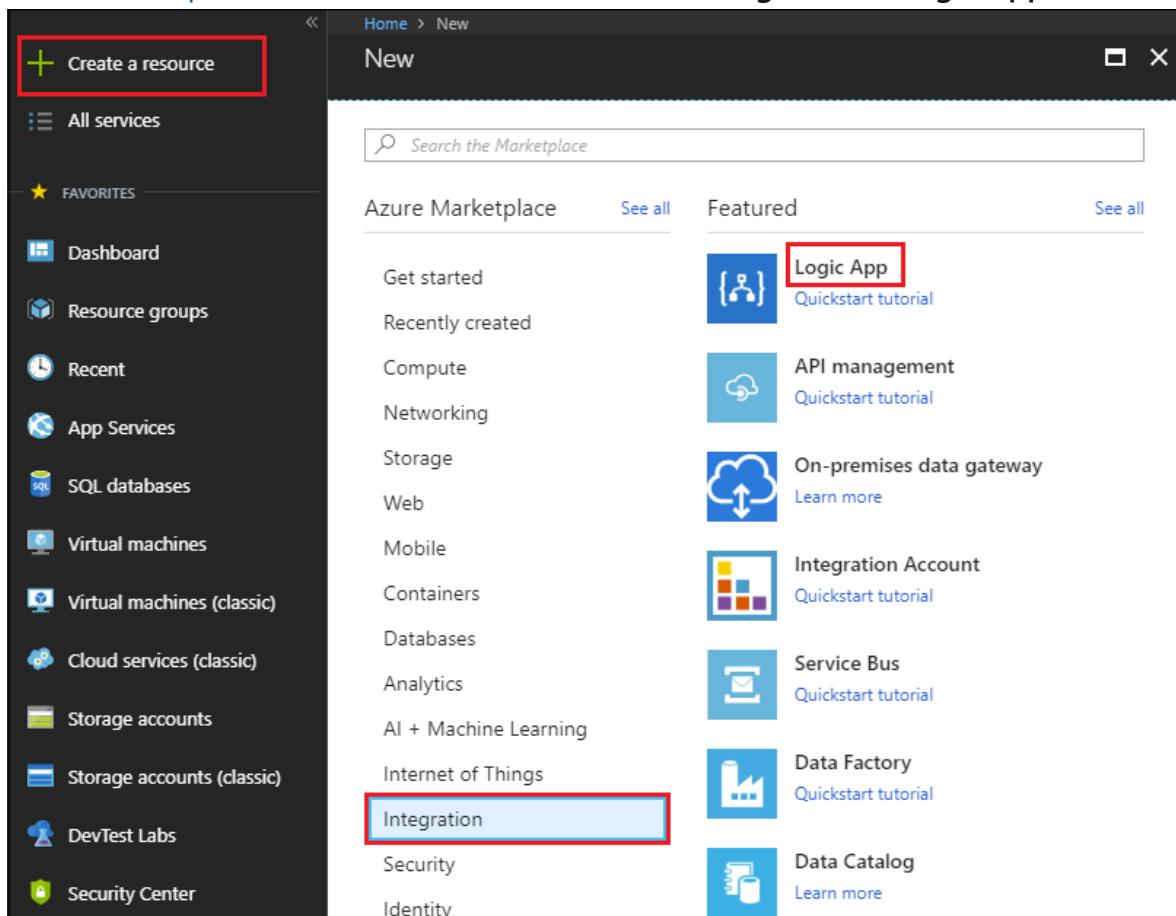
## Create the logic app

The logic app will perform several actions. The following list provides a high-level set of actions that the logic app will perform:

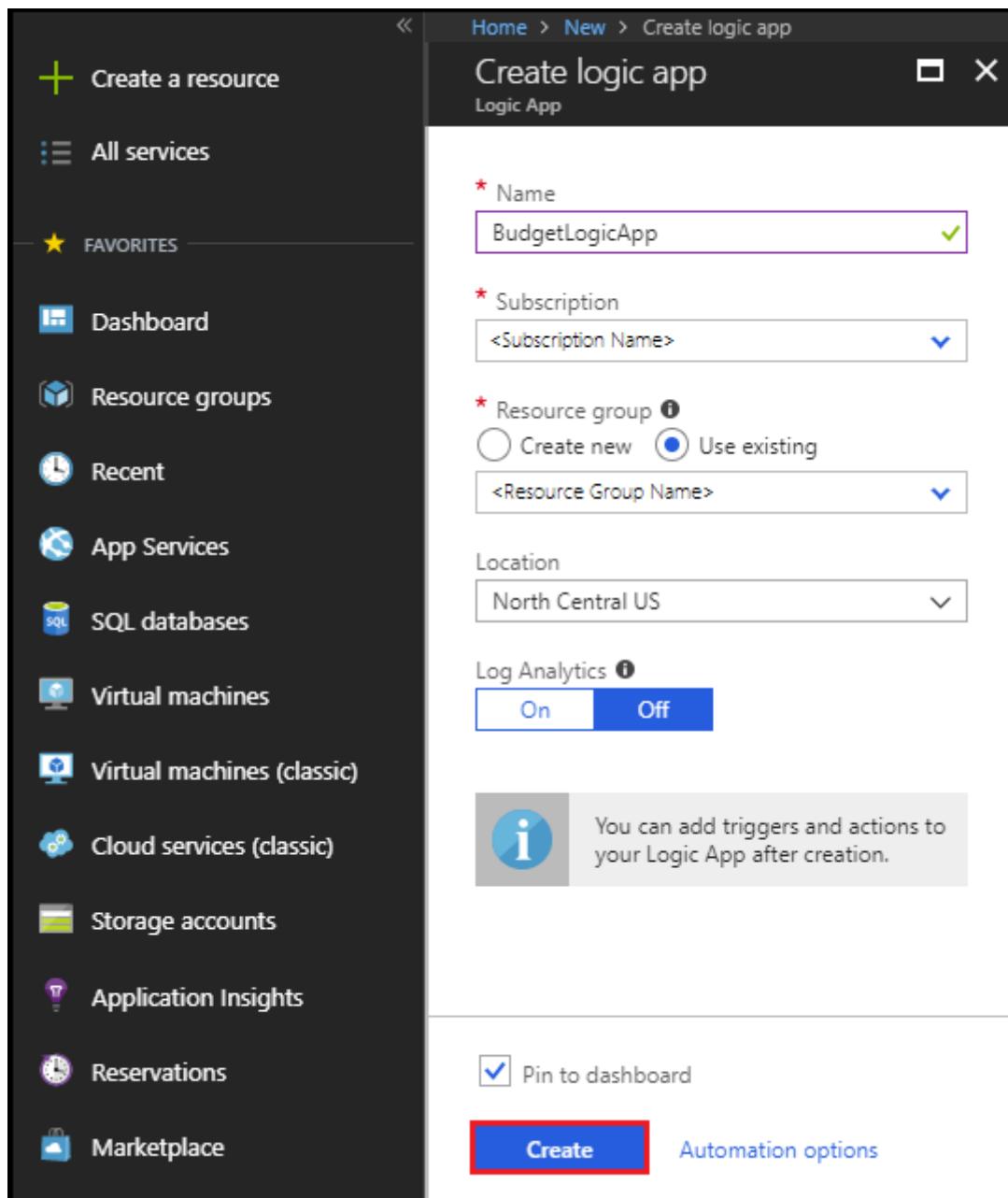
- Recognizes when an HTTP request is received
- Parse the passed in JSON data to determine the threshold value that has been reached
- Use a conditional statement to check whether the threshold amount has reached 80% or more of the budget range, but not greater than or equal to 100%.
  - If this threshold amount has been reached, send an HTTP POST using the webhook named **Optional**. This action will shut down the VMs in the "Optional" group.
- Use a conditional statement to check whether the threshold amount has reached or exceeded 100% of the budget value.
  - If the threshold amount has been reached, send an HTTP POST using the webhook named **Complete**. This action will shut down all remaining VMs.

The following steps are needed to create the logic app that will perform the above steps:

1. In the [Azure portal](#), select **Create a resource > Integration > Logic App**.



2. In the **Create logic app** area, provide the details need to create your logic app, select **Pin to dashboard**, and select **Create**.

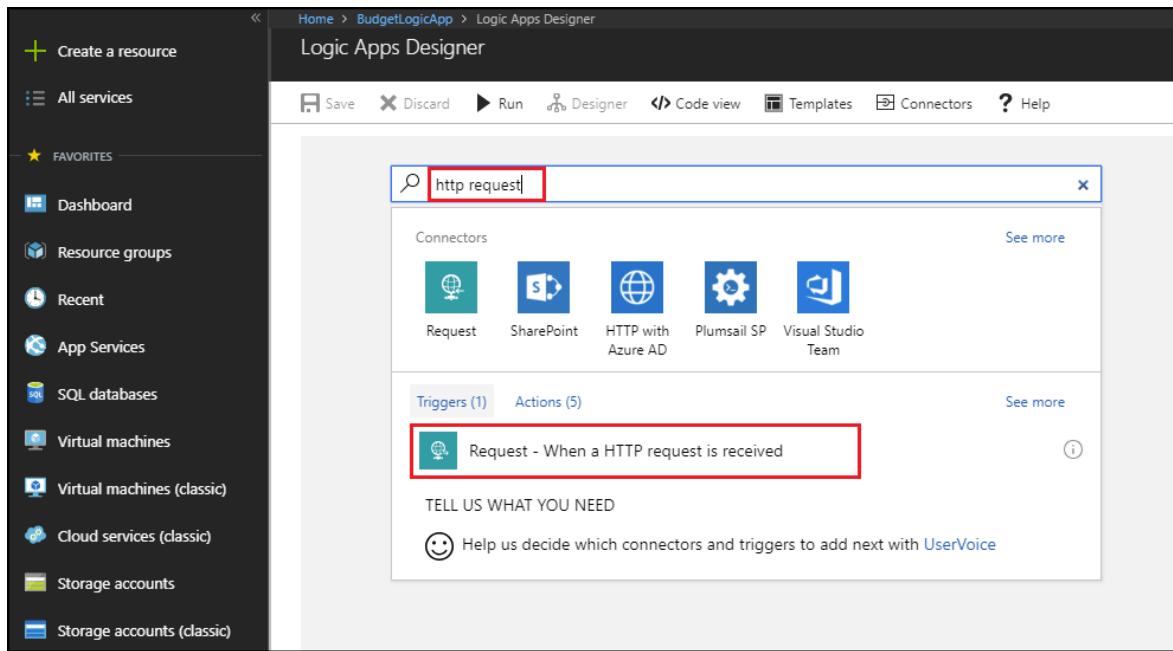


After Azure deploys your logic app, the **Logic Apps Designer** opens and shows an area with an introduction video and commonly used triggers.

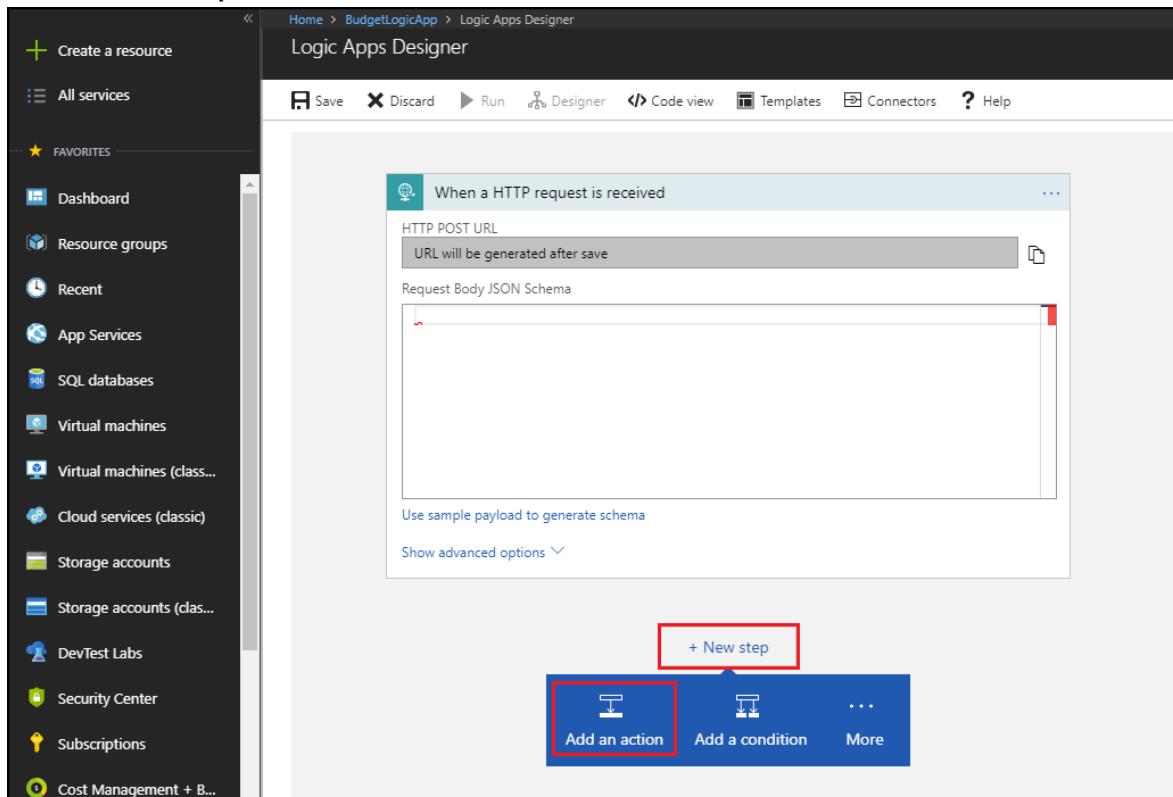
## Add a trigger

Every logic app must start with a trigger, which fires when a specific event happens or when a specific condition is met. Each time the trigger fires, the Logic Apps engine creates a logic app instance that starts and runs your workflow. Actions are all the steps that happen after the trigger.

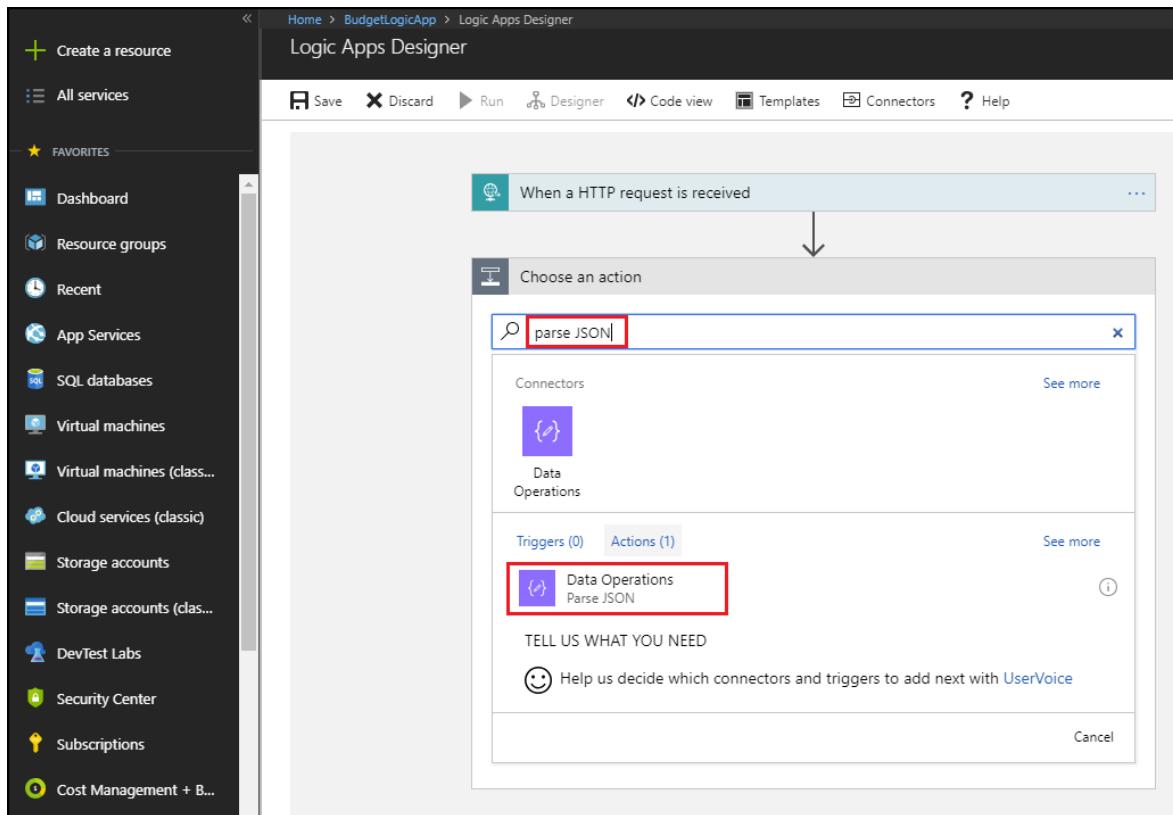
1. Under **Templates** of the **Logic Apps Designer** area, choose **Blank Logic App**.
2. Add a **trigger** by entering "http request" in the **Logic Apps Designer** search box to find and select the trigger named **Request – When an HTTP request is received**.



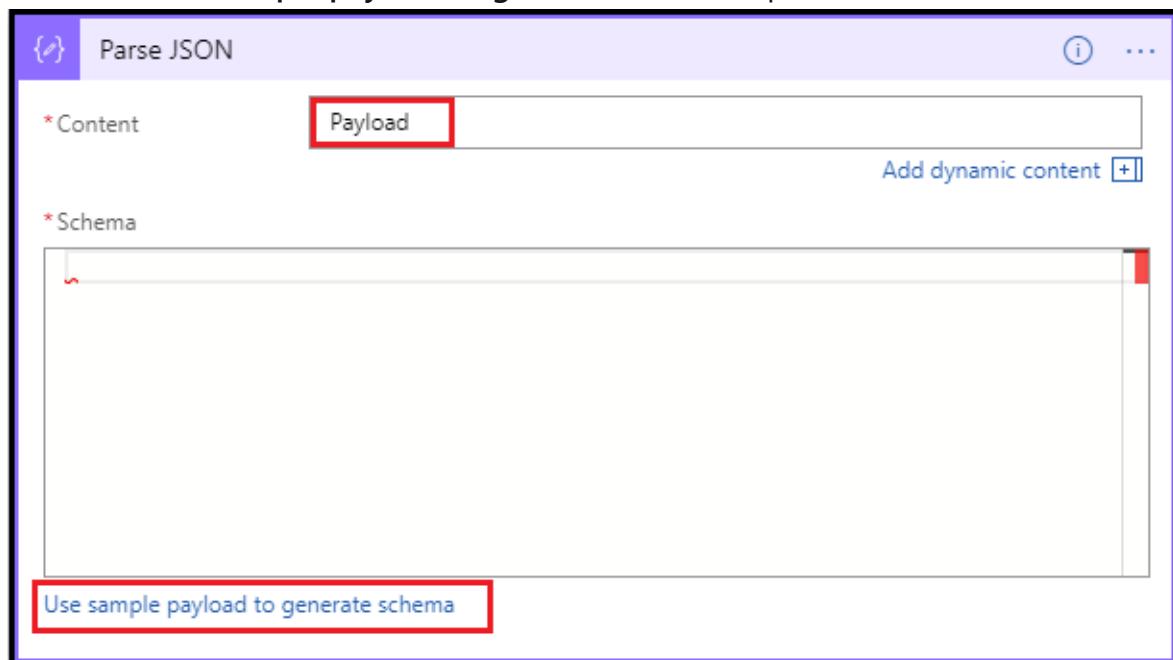
### 3. Select New step > Add an action.



### 4. Search for "parse JSON" in the Logic Apps Designer search box to find and select the Data Operations - Parse JSON action.



5. Enter "Payload" as the **Content** name for the Parse JSON payload or use the "Body" tag from dynamic content.
6. Select the **Use sample payload to generate schema** option in the Parse JSON box.



7. Paste the following JSON sample payload into the textbox:

```
{"schemaId": "AIP Budget Notification", "data": {"SubscriptionName": "CCM - Microsoft Azure Enterprise - 1", "SubscriptionId": "", "SpendingAmount": "100", "BudgetStartDate": "6/1/2018", "Budget": "50", "Unit": "USD", "BudgetCreator": "email@contoso.com", "BudgetName": "BudgetName", "BudgetType": "Cost", "ResourceGroup": "", "NotificationThresholdAmount": "0.8"}}
```

The text 'The' is also present at the end of the JSON block.

textbox will appear as:



8. Select Done.

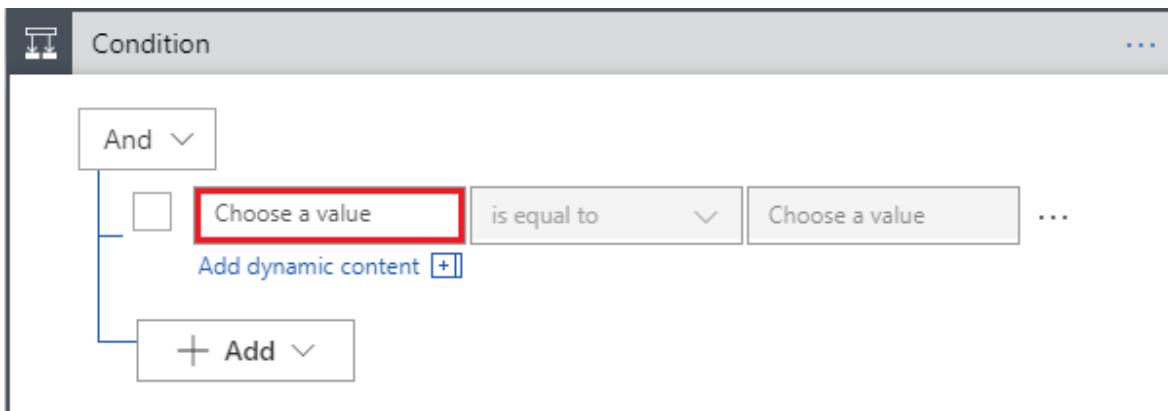
## Add the first conditional action

Use a conditional statement to check whether the threshold amount has reached 80% or more of the budget range, but not greater than or equal to 100%. If this threshold amount has been reached, send an HTTP POST using the webhook named **Optional**. This action will shut down the VMs in the **Optional** group.

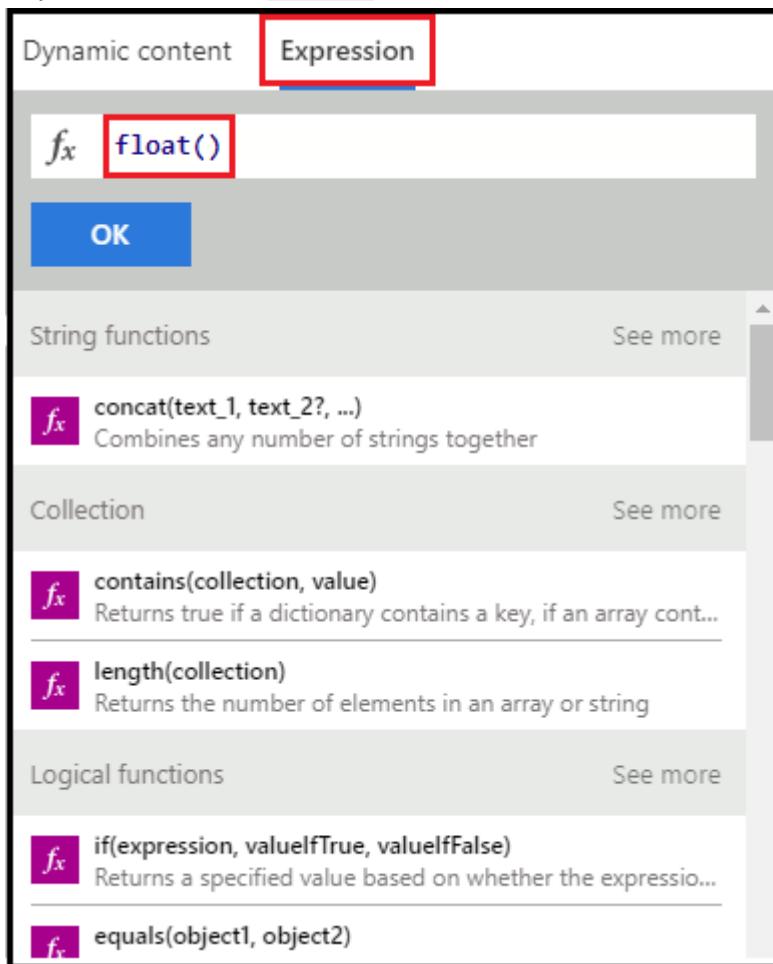
1. Select New step > Add a condition.



2. In the **Condition** box, select the textbox containing `Choose a value` to display a list of available values.



3. Select **Expression** at the top of the list and enter the following expression in the expression editor: `float()`



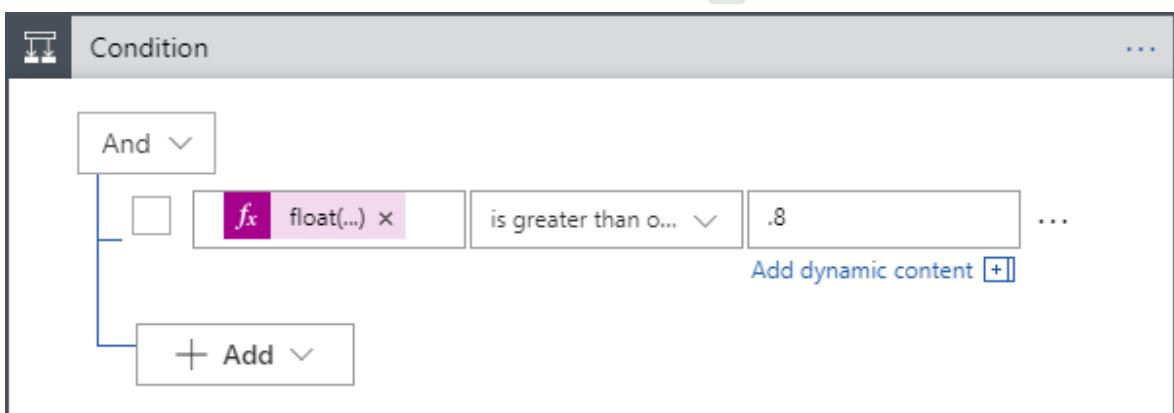
4. Select **Dynamic content**, place the cursor inside the parenthesis (), and select **NotificationThresholdAmount** from the list to populate the complete expression. The expression will be:

```
float(body('Parse_JSON')?['data']?['NotificationThresholdAmount'])
```

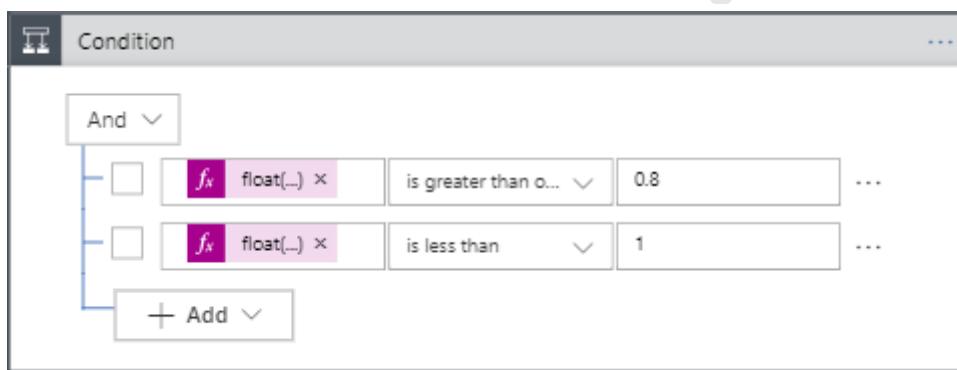
5. Select **OK** to set the expression.

6. Select **is greater than or equal to** in the dropdown box of the **Condition**.

7. In the **Choose a value** box of the condition, enter `.8`.



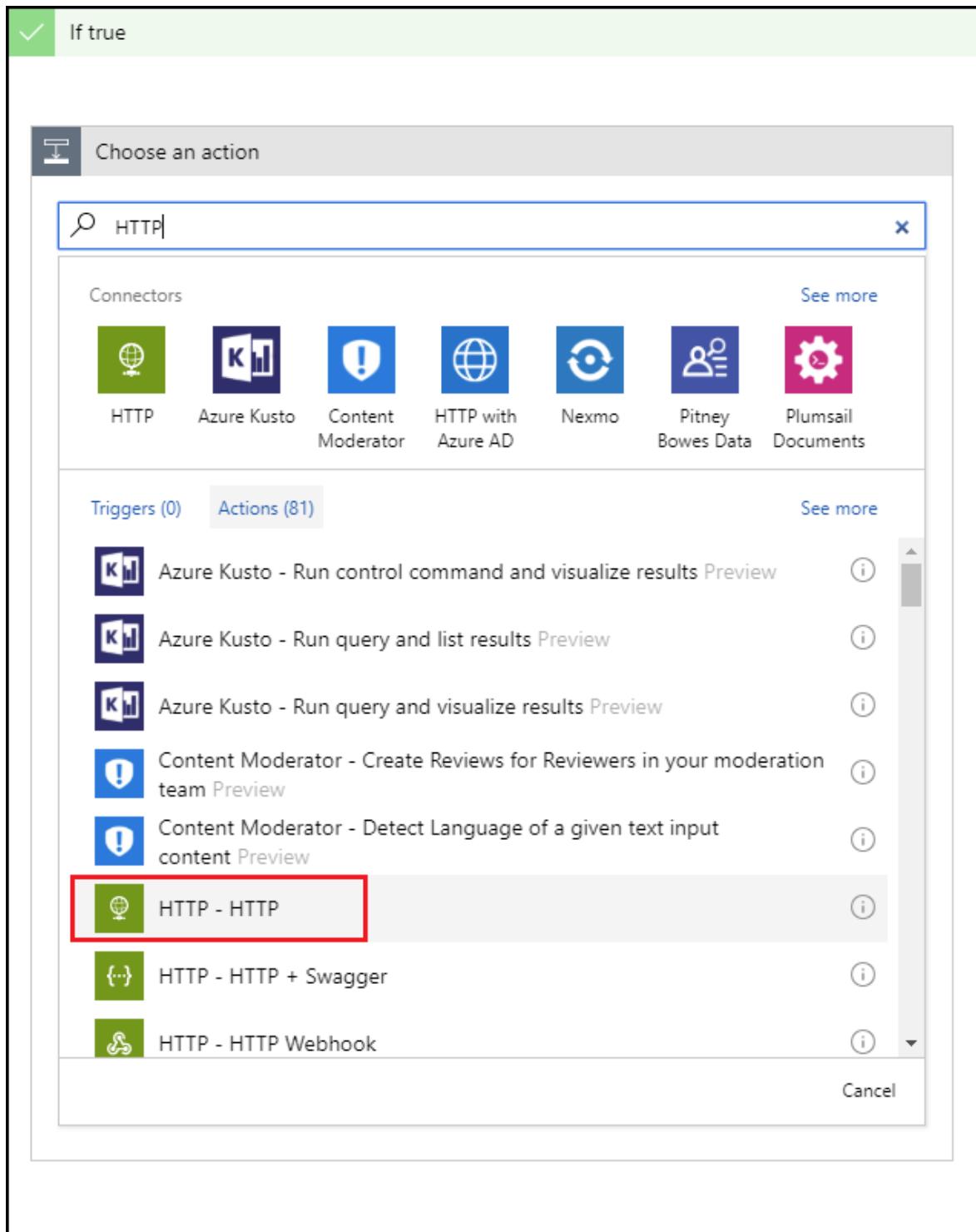
8. Select **Add > Add row** within the Condition box to add an additional part of the condition.
9. In the **Condition** box, select the textbox containing **Choose a value**.
10. Select **Expression** at the top of the list and enter the following expression in the expression editor: `float()`
11. Select **Dynamic content**, place the cursor inside the parenthesis (), and select **NotificationThresholdAmount** from the list to populate the complete expression.
12. Select **OK** to set the expression.
13. Select **is less than** in the dropdown box of the **Condition**.
14. In the **Choose a value** box of the condition, enter **1**.



15. In the **If true** box, select **Add an action**. You'll add an HTTP POST action that will shut down optional VMs.



16. Enter **HTTP** to search for the HTTP action and select the **HTTP – HTTP action**.



17. Select **Post** for the **Method** value.

18. Enter the URL for the webhook named **Optional** that you created earlier in this tutorial as the **Uri** value.

**HTTP**

* Method	post
* Uri	<webhook from Azure authomation>
Headers	Enter key   Enter value
Body	Enter request content

Show advanced options ▾

19. Select **Add an action** in the **If true** box. You'll add an email action that will send an email notifying the recipient that the optional VMs have been shut down.

20. Search for "send email" and select a *send email* action based on the email service you use.

Choose an action

x

Connectors

Office 365 Outlook	AWeber	Benchmark Email	FreshBooks	Gmail	MailChimp	Mandrill
--------------------	--------	-----------------	------------	-------	-----------	----------

Triggers (7) Actions (47) See more

- Office 365 Outlook - Send an email from a shared mailbox Preview
- Office 365 Outlook - Send email with options
- Outlook.com - Send an email
- Outlook.com - Send approval email
- Outlook.com - Delete email
- Outlook.com - Flag email Preview
- Outlook.com - Forward an email Preview
- Outlook.com - Get attachment

Cancel

For personal Microsoft accounts, select **Outlook.com**. For Azure work or school accounts, select **Office 365 Outlook**. If you don't already have a connection, you're asked to sign in to your email account. Logic Apps creates a connection to your email account. You'll need to allow the Logic App to access your email information.

 Microsoft

 Let this app access your info?

logic-apis-northcentralus.consent.azure-apim.net

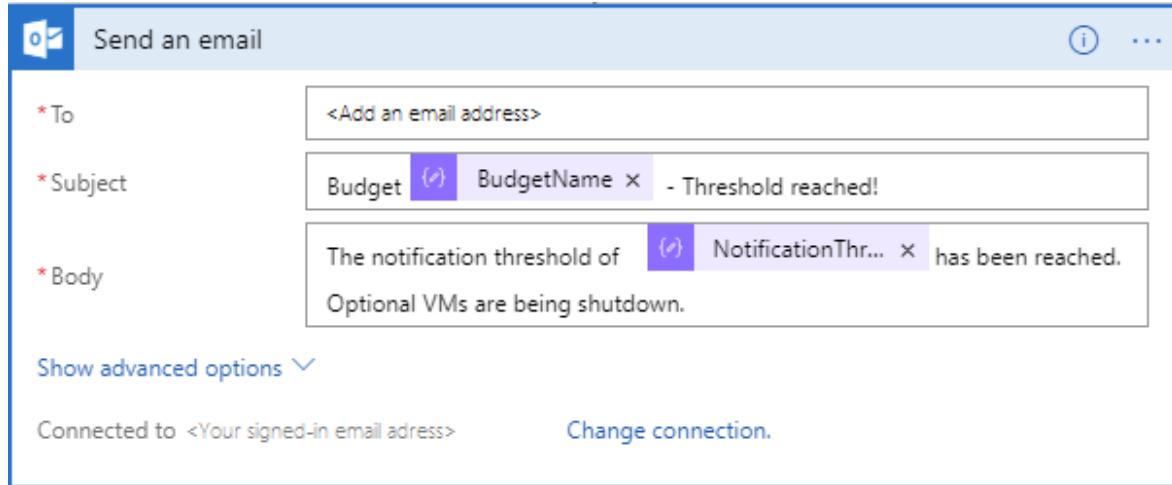
**Azure Logic Apps (North Central US) needs your permission to:**

-  **Read and write access to your mail**  
Azure Logic Apps (North Central US) will be able to read, update, create and delete email in your mailbox. Does not include permission to send mail.
-  **Send mail as you**  
Azure Logic Apps (North Central US) will be able to send mail as you.
-  **Have full access of your contacts**  
Azure Logic Apps (North Central US) will be able to read, update, create and delete contacts in your contact folders.
-  **Have full access to your calendars**  
Azure Logic Apps (North Central US) will be able to read, update, create and delete events in your calendars.
-  **Access your info anytime**  
Azure Logic Apps (North Central US) will be able to see and update your info, even when you're not using this app.

Accepting these permissions means that you allow this app to use your data as specified in their [terms of service](#) and [privacy statement](#). You can change these permissions at <https://microsoft.com/consent>. [Show details](#)

[Sign out](#)   [© 2018 Microsoft](#)   [Terms of Use](#)   [Privacy & Cookies](#)

21. Add the **To**, **Subject**, and **Body** text for the email that notifies the recipient that the optional VMs have been shut down. Use the **BudgetName** and the **NotificationThresholdAmount** dynamic content to populate the subject and body fields.



## Add the second conditional action

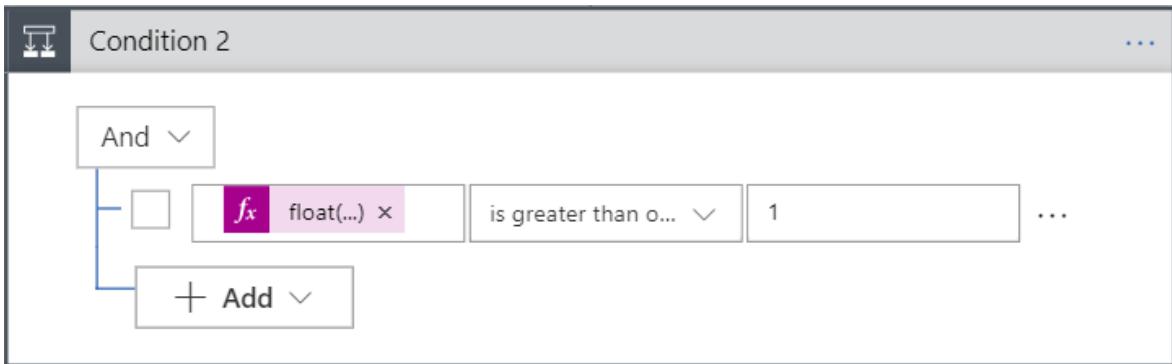
Use a conditional statement to check whether the threshold amount has reached or exceeded 100% of the budget value. If the threshold amount has been reached, send an HTTP POST using the webhook named **Complete**. This action will shut down all remaining VMs.

1. Select **New step > Add a Condition**.



2. In the **Condition** box, select the textbox containing `Choose a value` to display a list of available values.
3. Select **Expression** at the top of the list and enter the following expression in the expression editor: `float()`
4. Select **Dynamic content**, place the cursor inside the parenthesis (), and select **NotificationThresholdAmount** from the list to populate the complete expression. The expression will resemble:  
`float(body('Parse_JSON')?['data']?['NotificationThresholdAmount'])`
5. Select **OK** to set the expression.
6. Select **is greater than or equal to** in the dropdown box of the **Condition**.

7. In the Choose a value box for the condition, enter 1.



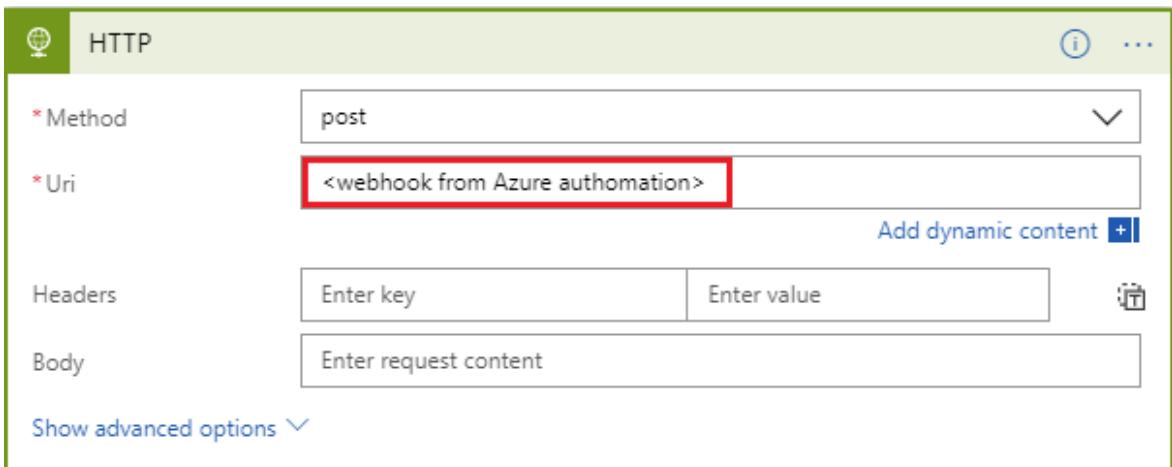
8. In the If true box, select Add an action. You'll add an HTTP POST action that will shut down all the remaining VMs.



9. Enter **HTTP** to search for the HTTP action and select the **HTTP – HTTP action**.

10. Select **Post** as the **Method** value.

11. Enter the URL for the webhook named **Complete** that you created earlier in this tutorial as the **Uri** value.

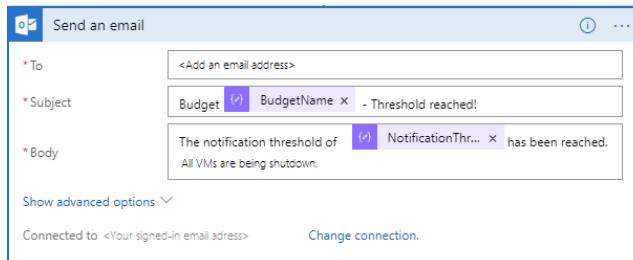


12. Select **Add an action** in the **If true** box. You'll add an email action that will send an email notifying the recipient that the remaining VMs have been shut down.

13. Search for "send email" and select a *send email* action based on the email service you use.

14. Add the **To**, **Subject**, and **Body** text for the email that notifies the recipient that the optional VMs have been shut down. Use the **BudgetName** and the **NotificationThresholdAmount** dynamic content to populate the subject and body

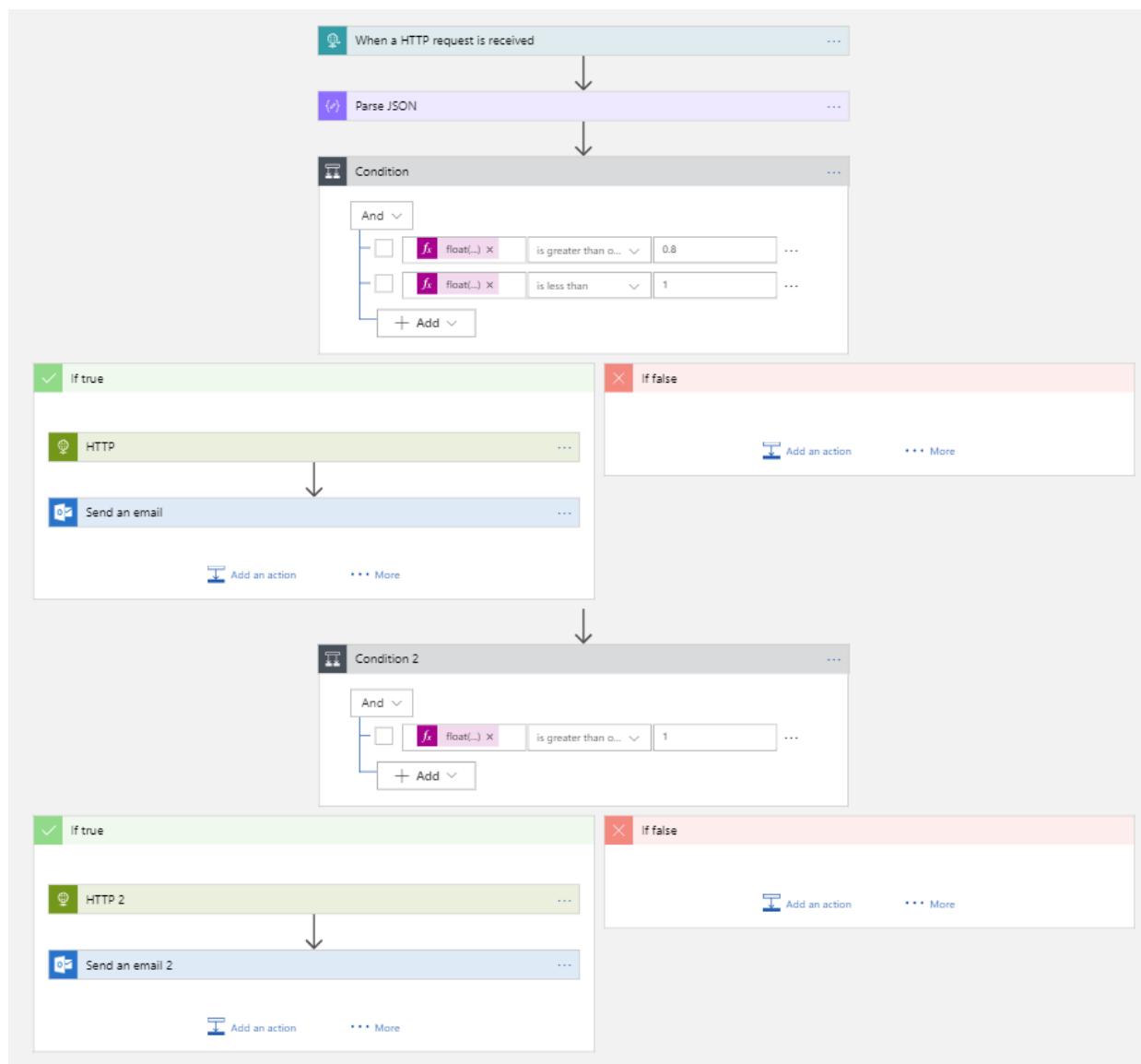
fields.



15. Select **Save** at the top of the Logic App Designer area.

## Logic App summary

Here's what your Logic App looks like once you're done. In the most basic of scenarios where you don't need any threshold-based orchestration, you could directly call the automation script from **Monitor** and skip the **Logic App** step.



When you saved your logic app, a URL was generated that you'll be able to call. You'll use this URL in the next section of this tutorial.

# Create an Azure Monitor Action Group

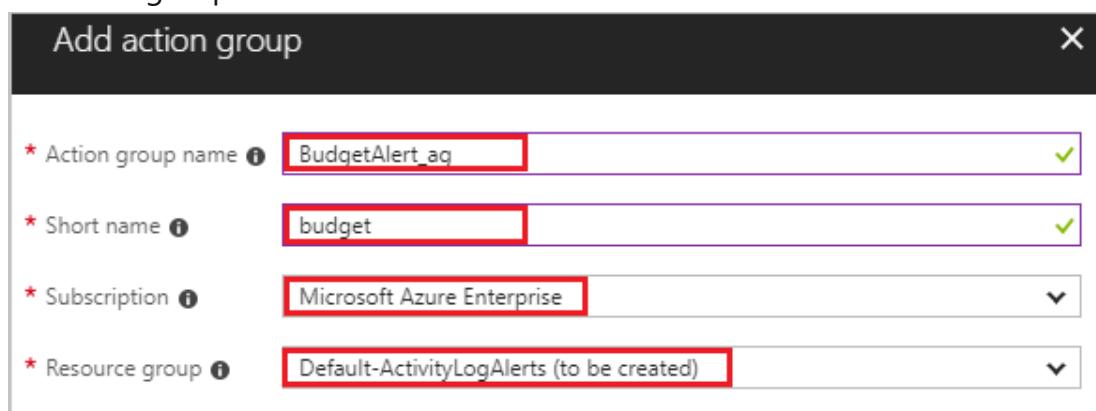
An action group is a collection of notification preferences that you define. When an alert is triggered, a specific action group can receive the alert by being notified. An Azure alert proactively raises a notification based on specific conditions and provides the opportunity to take action. An alert can use data from multiple sources, including metrics and logs.

Action groups are the only endpoint that you'll integrate with your budget. You can set up notifications in a number of channels, but for this scenario you'll focus on the Logic App you created earlier in this tutorial.

## Create an action group in Azure Monitor

When you create the action group, you'll point to the Logic App that you created earlier in this tutorial.

1. If you are not already signed-in to the [Azure portal](#), sign in and select **All services > Monitor**.
2. Select **Alerts** then select **Manage actions**.
3. Select **Add an action group** from the **Action groups** area.
4. Add and verify the following items:
  - Action group name
  - Short name
  - Subscription
  - Resource group



5. Within the **Add action group** pane, add a **LogicApp** action. Name the action **Budget-BudgetLA**. In the **Logic App** pane, select the **Subscription** and the **Resource group**. Then, select the **Logic app** that you created earlier in this tutorial.
6. Select **OK** to set the Logic App. Then, select **OK** in the **Add action group** pane to create the action group.

You're done with all the supporting components needed to effectively orchestrate your budget. Now all you need to do is create the budget and configure it to use the action group you created.

## Create the budget

You can create a budget in the Azure portal using the [Budget feature](#) in Cost Management. Or, you can create a budget using REST APIs, PowerShell cmdlets, or use the CLI. The following procedure uses the REST API. Before calling the REST API, you'll need an authorization token. To create an authorization token, you can use the [ARMClient](#) project. The **ARMClient** allows you to authenticate yourself to the Azure Resource Manager and get a token to call the APIs.

## Create an authentication token

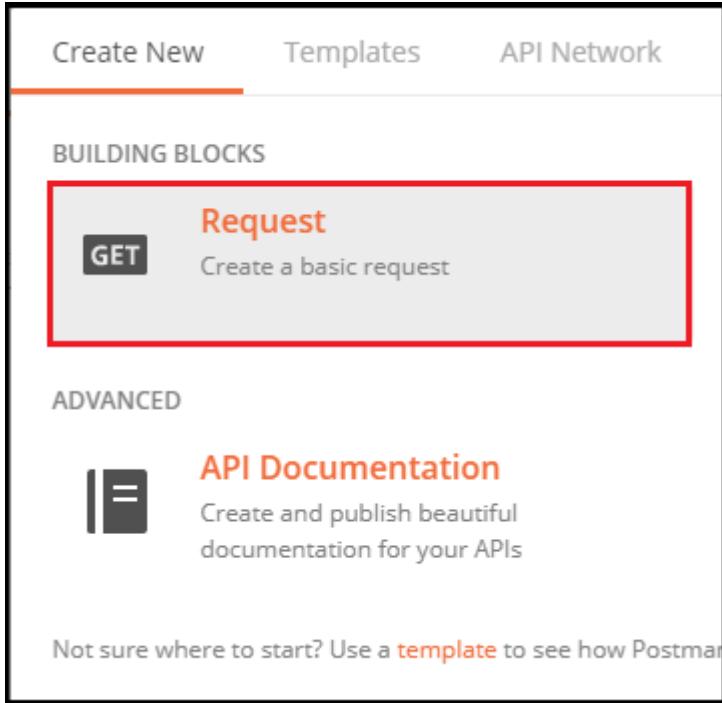
1. Navigate to the [ARMClient](#) project on GitHub.
2. Clone the repo to get a local copy.
3. Open the project in Visual Studio and build it.
4. Once the build is successful, the executable should be in the `\bin\debug` folder.
5. Run the ARMClient. Open a command prompt and navigate to the `\bin\debug` folder from the project root.
6. To sign in and authenticate, enter the following command at the command prompt:  
`ARMClient login prod`
7. Copy the **subscription guid** from the output.
8. To copy an authorization token to your clipboard, enter the following command at the command prompt, but sure to use the copied subscription ID from the step above:  
`ARMClient token <subscription GUID from previous step>`

Once you have completed the step above, you'll see:  
**Token copied to clipboard successfully.**
9. Save the token to be used for steps in the next section of this tutorial.

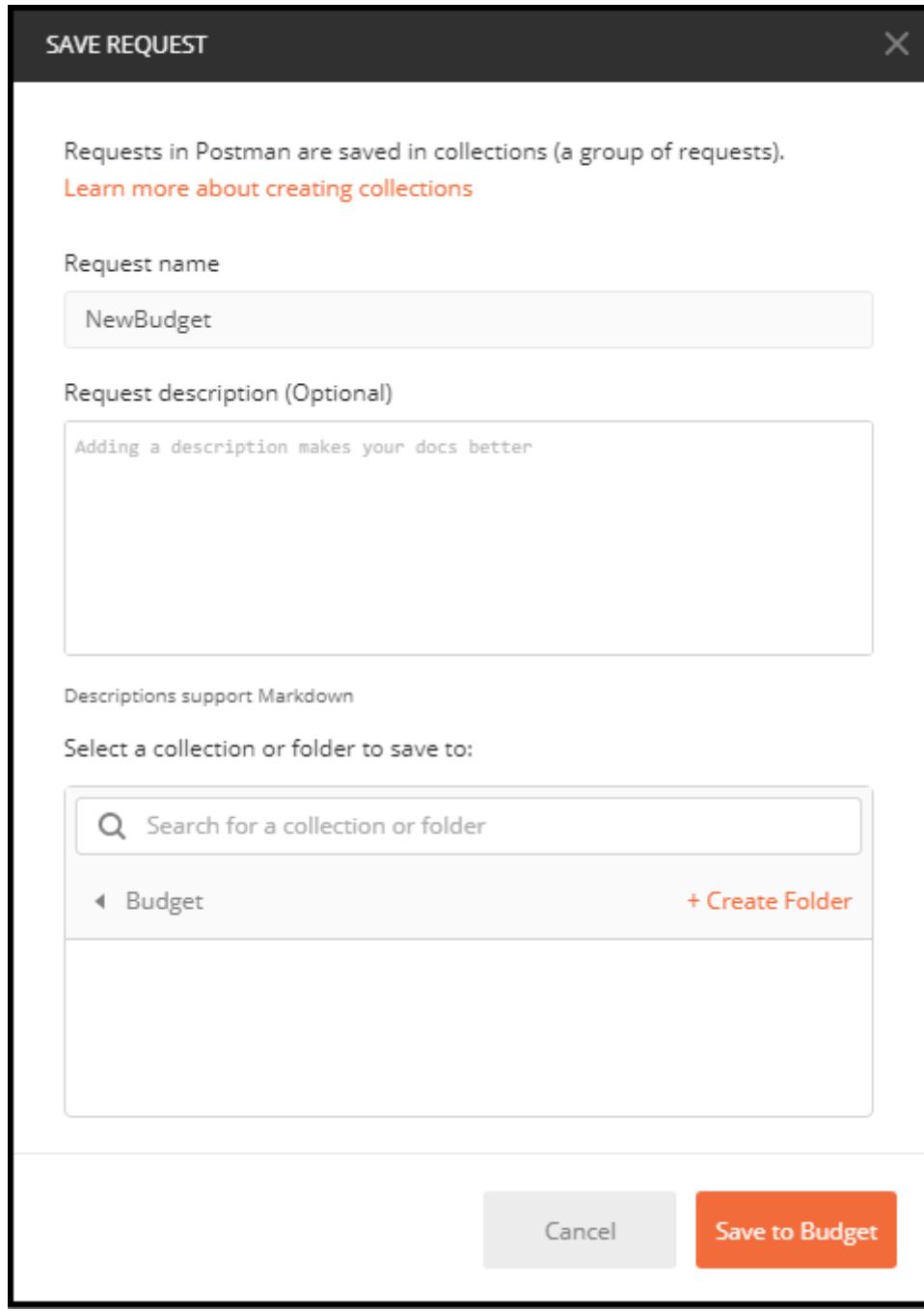
## Create the Budget

Next, you'll configure **Postman** to create a budget by calling the Azure Consumption REST APIs. Postman is an API Development environment. You'll import environment and collection files into Postman. The collection contains grouped definitions of HTTP requests that call Azure Consumption REST APIs. The environment file contains variables that are used by the collection.

1. Download and open the [Postman REST client](#) to execute the REST APIs.
2. In Postman, create a new request.



3. Save the new request as a collection, so that the new request has nothing on it.



4. Change the request from a `Get` to a `Put` action.

5. Modify the following URL by replacing `{subscriptionId}` with the **Subscription ID** that you used in the previous section of this tutorial. Also, modify the URL to include "SampleBudget" as the value for `{budgetName}`:

```
https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.Consumption/budgets/{budgetName}?api-version=2018-03-31
```

6. Select the **Headers** tab within Postman.

7. Add a new **Key** named "Authorization".

8. Set the **Value** to the token that was created using the ArmClient at the end of the last section.

9. Select **Body** tab within Postman.

10. Select the **raw** button option.

11. In the textbox, paste in the below sample budget definition, however you must replace the `subscriptionID`, `resourcegroupname`, and `actiongroupname` parameters with your subscription ID, a unique name for your resource group, and the action group name you created in both the URL and the request body:

```
{
 "properties": {
 "category": "Cost",
 "amount": 100.00,
 "timeGrain": "Monthly",
 "timePeriod": {
 "startDate": "2018-06-01T00:00:00Z",
 "endDate": "2018-10-31T00:00:00Z"
 },
 "filters": {},
 "notifications": {
 "Actual_GreaterThan_80_Percent": {
 "enabled": true,
 "operator": "GreaterThan",
 "threshold": 80,
 "contactEmails": [],
 "contactRoles": [],
 "contactGroups": [
 "/subscriptions/{subscriptionid}/resourceGroups/{resourcegroupname}/providers/microsoft.insights/actionGroups/{actiongroupname}"
]
 },
 "Actual_EqualTo_100_Percent": {
 "operator": "EqualTo",
 "threshold": 100,
 "contactGroups": [
 "/subscriptions/{subscriptionid}/resourceGroups/{resourcegroupname}/providers/microsoft.insights/actionGroups/{actiongroupname}"
]
 }
 }
}
```

12. Press **Send** to send the request.

You now have all the pieces you need to call the [budgets API](#). The budgets API reference has additional details on the specific requests, including:

- **budgetName** - Multiple budgets are supported. Budget names must be unique.
- **category** - Must be either **Cost** or **Usage**. The API supports both cost and usage budgets.
- **timeGrain** - A monthly, quarterly, or yearly budget. The amount resets at the end of the period.
- **filters** - Filters allow you to narrow the budget to a specific set of resources within the selected scope. For example, a filter could be a collection of resource groups for a subscription level budget.
- **notifications** – Determines the notification details and thresholds. You can set up multiple thresholds and provide an email address or an action group to receive a notification.

## Summary

By using this tutorial, you learned:

- How to create an Azure Automation Runbook to stop VMs.
- How to create an Azure Logic App that is triggered based on the budget threshold values and call the related runbook with the right parameters.
- How to create an Azure Monitor Action Group that was configured to trigger the Azure Logic App when the budget threshold is met.
- How to create the budget with the desired thresholds and wire it to the action group.

You now have a fully functional budget for your subscription that will shut down your VMs when you reach your configured budget thresholds.

## Next steps

- For more information about Azure billing scenarios, see [Billing and cost management automation scenarios](#).

# APIs for Azure reservation automation

Article • 12/07/2022

Use Azure APIs to programmatically get information for your organization about Azure service or software reservations.

## Find reservation plans to buy

Use the Reservation recommendation API to get recommendations on which reservations plan to buy based on your organization's usage. For more information, see [Reservation Recommendations](#).

You can also analyze your resource usage by using the Consumption API Usage Detail. For more information, see [Usage Details - List For Billing Period By Billing Account](#). The Azure resources that you use consistently are usually the best candidate for a reservation.

## Buy a reservation

You can purchase Azure reservations and software plans programmatically by using REST APIs. To learn more, see [Reservation Order - Purchase API](#).

Here's a sample request to purchase by using the REST API:

HTTP

PUT

```
https://management.azure.com/providers/Microsoft.Capacity/reservationOrders/
<GUID>?api-version=2019-04-01
```

Request body:

JSON

```
{
 "sku": {
 "name": "standard_D1"
 },
 "location": "westus",
 "properties": {
 "reservedResourceType": "VirtualMachines",
 "billingScopeId": "/subscriptions/ed3a1871-612d-abcd-a849-c2542a68be83",
 "term": "P1Y",
 "quantity": "1",
 }
}
```

```
 "displayName": "TestReservationOrder",
 "appliedScopes": null,
 "appliedScopeType": "Shared",
 "reservedResourceProperties": {
 "instanceFlexibility": "On"
 }
}
```

You can also buy a reservation in the Azure portal. For more information, see the following articles:

Service plans:

- [Virtual machine](#)
- [Azure Cosmos DB](#)
- [SQL Database](#)

Software plans:

- [SUSE Linux software](#)

## Get reservations

If you're an Azure customer with an Enterprise Agreement (EA customer), you can get the reservations your organization bought by using the [Reservation Transactions - List](#). For other subscriptions, get the list of reservations you bought and have permissions to view by using the API [Reservation Order - List](#). By default, the account owner or person that bought the reservation has permissions to view the reservation.

## See reservation usage

If you're an EA customer, you can programmatically view how the reservations in your organization are being used. For more information, see [Reservation Transactions - List](#). For other subscriptions, use the API [Reservations Summaries - List By Reservation Order And Reservation](#).

If you find that your organization's reservations are being under-used:

- Make sure the virtual machines that your organization creates match the VM size that's on the reservation.
- Make sure instance size flexibility is on. For more information, see [Manage reservations - Change optimize setting for Reserved VM Instances](#).

- Change the scope of reservation to shared so that it applies more broadly. For more information, see [Manage reservations - Change the scope for a reservation](#).
- Exchange the unused quantity. For more information, see [Manage reservations](#).

## Give access to reservations

Get the list of all reservations that a user has access to by using the [Reservation - Operation - List API](#). To give access to a reservation programmatically, see one of the following articles:

- [Add or remove Azure role assignments using the REST API](#)
- [Add or remove Azure role assignments using Azure PowerShell](#)
- [Add or remove Azure role assignments using Azure CLI](#)

## Split or merge reservation

After you buy more than one resource instance within a reservation, you may want to assign instances within that reservation to different subscriptions. You can change the reservation scope so that it applies to all subscriptions within the same billing context. But for cost management or budgeting purposes, you may want to keep the scope as "single subscription" and assign reservation instances to a specific subscription.

To split a reservation, use the API [Reservation - Split](#). You can also split a reservation by using PowerShell. For more information, see [Manage reservations - Split reservation into two reservations](#).

To merge two reservations into one reservation, use the API [Reservation - Merge](#).

## Change scope for a reservation

The scope of a reservation can be single subscription, single resource group or all subscriptions in your billing context. If you set the scope to single subscription or single resource group, the reservation is matched to running resources in the selected subscription. If you delete or move the subscription or the resource group, the reservation will not be utilized. If you set the scope to shared, Azure matches the reservation to resources that run in all the subscriptions within the billing context. The billing context is dependent on the subscription you used to buy the reservation. You can select the scope at purchase or change it anytime after purchase. For more information, see [Manage Reservations - Change the scope](#).

To change the scope programmatically, use the API [Reservation - Update](#).

## Learn more

- What are reservations for Azure
- Understand how the VM reservation discount is applied
- Understand how the SUSE Linux Enterprise software plan discount is applied
- Understand how other reservation discounts are applied
- Understand reservation usage for your Pay-As-You-Go subscription
- Understand reservation usage for your Enterprise enrollment
- Windows software costs not included with reservations
- Azure Reservations in Partner Center Cloud Solution Provider (CSP) program

# Migrate from Azure Enterprise Reporting to Microsoft Cost Management APIs overview

Article • 12/19/2022

This article helps developers that have built custom solutions using the [Azure Enterprise Reporting APIs](#) to migrate to Microsoft Cost Management APIs. Service principal support is available in the newer Cost Management APIs and they are still actively being developed. Consider migrating to them instead of using the older Azure Enterprise Reporting APIs. The older APIs are being deprecated. This article helps you understand the differences between the Azure Enterprise Reporting APIs and the Cost Management APIs, what to expect when you migrate to the Cost Management APIs, and the new capabilities that are available with the Cost Management APIs.

## API differences

The following information describes the differences between the older Azure Enterprise Reporting APIs and the newer Cost Management APIs.

Use	Azure Enterprise Reporting APIs	Microsoft Cost Management APIs
Authentication	API key provisioned in the Enterprise Agreement (EA) portal	Azure Active Directory (Azure AD) Authentication using user tokens or service principals. Service principals take the place of API keys.
Scopes and permissions	All requests are at the enrollment scope. API Key permission assignments will determine whether data for the entire enrollment, a department, or a specific account is returned. No user authentication.	Users or service principals are assigned access to the enrollment, department, or account scope.
URI Endpoint	<a href="https://consumption.azure.com">https://consumption.azure.com</a>	<a href="https://management.azure.com">https://management.azure.com</a>
Development status	In maintenance mode. On the path to deprecation.	In active development
Available APIs	Limited to what's currently available	Equivalent APIs are available to replace each EA API. Additional <a href="#">Cost Management APIs</a> are also available, including: <ul style="list-style-type: none"><li>- Budgets</li><li>- Alerts</li><li>- Exports</li></ul>

## Migration checklist

- Familiarize yourself with the [Azure Resource Manager REST APIs](#).
- Determine which Enterprise Reporting APIs you use and see which Cost Management APIs to move to at [Migrate from Azure Enterprise Reporting to Microsoft Cost Management APIs](#).
- Configure service authorization and authentication for the Cost Management APIs. For more information, see [Assign permission to ACM APIs](#).

- Test the APIs and then update any programming code to replace Enterprise Reporting API calls with Cost Management API calls.
- Update error handling to use new error codes. Some considerations include:
  - Cost Management APIs have a timeout period of 60 seconds.
  - Cost Management APIs have rate limiting in place. This results in a `429 throttling error` if rates are exceeded. Build your solutions so that you don't make too many API calls in a short time period.
- Review the other Cost Management APIs available through Azure Resource Manager and assess for use later. For more information, see [Migrate from Azure Enterprise Reporting to Microsoft Cost Management APIs](#).

## Enterprise Reporting API mapping to new Cost Management APIs

Use the following information to identify the Enterprise Reporting APIs that you currently use and the replacement Cost Management API to use instead.

Scenario	Enterprise Reporting APIs	Cost Management APIs
Migrate from EA Usage Details APIs	/usagedetails/download /usagedetails/submit /usagedetails /usagedetailsbycustomdate	Use <a href="#">Microsoft.CostManagement/Exports</a> for all recurring data ingestion workloads. Use the <a href="#">Cost Details</a> report for small on-demand datasets.
Migrate from EA Balance Summary APIs	/balancesummary	<a href="#">Microsoft.Consumption/balances</a>
Migrate from EA Price Sheet APIs	/pricesheet	For negotiated prices, use <a href="#">Microsoft.Consumption/pricesheets/default</a> For retail prices, use <a href="#">Retail Prices API</a>
Migrate from EA Reserved Instance Usage Details API	/reservationdetails	<a href="#">Microsoft.CostManagement/generateReservationDetailsReport</a>
Migrate from EA Reserved Instance Usage Summary APIs	/reservationsummaries	<a href="#">Microsoft.Consumption/reservationSummaries</a>
Migrate from EA Reserved Instance Recommendations APIs	/SharedReservationRecommendations /SingleReservationRecommendations	<a href="#">Microsoft.Consumption/reservationRecommendations</a>
Migrate from EA Reserved Instance Charges APIs	/reservationcharges	<a href="#">Microsoft.Consumption/reservationTransactions</a>

## Use additional Cost Management APIs

After you've migrated to the Cost Management APIs for your existing reporting scenarios, you can use many other APIs, too. The APIs are also available through Azure Resource Manager and can be automated using service principal-based authentication. Here's a quick summary of the new capabilities that you can use.

- [Budgets](#) - Use to set thresholds to proactively monitor your costs, alert relevant stakeholders, and automate actions in response to threshold breaches.
- [Alerts](#) - Use to view alert information including, but not limited to, budget alerts, invoice alerts, credit alerts, and quota alerts.
- [Exports](#) - Use to schedule recurring data export of your charges to an Azure Storage account of your choice. It's the recommended solution for customers with a large Azure presence who want to analyze their data and use it in their own internal systems.

## Next steps

- Familiarize yourself with the [Azure Resource Manager REST APIs](#).
- If needed, determine which Enterprise Reporting APIs you use and see which Cost Management APIs to move to at [Migrate from Azure Enterprise Reporting to Microsoft Cost Management APIs](#).
- If you're not already using Azure Resource Manager APIs, [register your client app with Azure AD](#).
- If needed, update any of your programming code to use [Azure AD authentication](#) with your service principal.

# Migrate from EA Usage Details APIs

Article • 07/18/2022

EA customers who were previously using the Enterprise Reporting APIs behind the *consumption.azure.com* endpoint to obtain usage details and marketplace charges need to migrate to new and improved solutions. Instructions are outlined below along with contract differences between the old API and the new solutions.

The dataset is referred to as *cost details* instead of *usage details*.

## New solutions generally available

The following table provides a summary of the migration destinations that are available along with a summary of what to consider when choosing which solution is best for you.

Solution	Description	Considerations	Onboarding info
Exports	Recurring data dumps to storage on a schedule	<ul style="list-style-type: none"><li>- The most scalable solution for your workloads.</li><li>- Can be configured to use file partitioning for bigger datasets.</li><li>- Great for establishing and growing a cost dataset that can be integrated with your own queryable data stores.</li><li>- Requires access to a storage account that can hold the data.</li></ul>	<ul style="list-style-type: none"><li>- Configure in Azure portal</li><li>Automate Export creation with the API</li><li>- Export API Reference</li></ul>
Cost Details API	On demand download	<ul style="list-style-type: none"><li>- Useful for small cost datasets.</li><li>- Useful for scenarios when Exports to Azure storage aren't feasible due to security or manageability concerns.</li></ul>	<ul style="list-style-type: none"><li>- Get small cost datasets on demand</li><li>- Cost Details API</li></ul>

Generally we recommend using [Exports](#) if you have ongoing data ingestion needs and/or a large monthly cost details dataset. For more information, see [Ingest cost details data](#). If you need additional information to help you make a decision for your workload, see [Choose a cost details solution](#).

## Assign permissions to an SPN to call the APIs

If you're looking to call either the Exports or Cost Details APIs programmatically, you'll need to configure a Service Principal with the correct permission. For more information,

see [Assign permissions to ACM APIs](#).

## Avoid the Microsoft Consumption Usage Details API

The [Consumption Usage Details API](#) is another endpoint that currently supports EA customers. Don't migrate to this API. Migrate to either Exports or the Cost Details API, as outlined earlier in this document. The Consumption Usage Details API will be deprecated in the future and is located behind the endpoint below.

HTTP

GET

```
https://management.azure.com/{scope}/providers/Microsoft.Consumption/usageDetails?api-version=2021-10-01
```

This API is a synchronous endpoint and will be unable to scale as both your spending and the size of your month over month cost dataset increases. If you're currently using the Consumption Usage Details API, we recommend migrating off of it to either Exports or the Cost Details API as soon as possible. A formal deprecation announcement will be made at a future date and a timeline for retirement will be provided. To learn more about migrating away from Consumption Usage Details, see [Migrate from Consumption Usage Details API](#).

## Migration benefits

Our new solutions provide many benefits over the EA Reporting Usage Details APIs. Here's a summary:

- **Security and stability** - New solutions require Service Principal and/or user tokens in order to access data. They're more secure than the API keys that are used for authenticating to the EA Reporting APIs. Keys in these legacy APIs are valid for six months and can expose sensitive financial data if leaked. Additionally, if keys aren't renewed and integrated into workloads prior to their six month expiry data access is revoked. This breaks customer workloads.
- **Scalability** - The EA Reporting APIs aren't built to scale well as your Azure usage increases. The usage details dataset can get exceedingly large as you deploy more resources into the cloud. The new solutions are asynchronous and have extensive infrastructure enhancements behind them to ensure successful downloads for any size dataset.
- **Single dataset for all usage details** - Azure and Azure Marketplace usage details have been merged into one dataset in the new solutions. The single dataset

reduces the number of APIs that you need to call to see all your charges.

- **Purchase amortization** - Customers who purchase Reservations can see an Amortized view of their costs using the new solutions.
- **Schema consistency** - Each solution that is available provides files with matching fields. It allows you to easily move between solutions based on your scenario.
- **Cost Allocation integration** - Enterprise Agreement and Microsoft Customer Agreement customers can use the new solution to view charges in relation to the cost allocation rules that they've configured. For more information about cost allocation, see [Allocate costs](#).
- **Go forward improvements** - The new solutions are being actively developed moving forward. They'll receive all new features as they're released.

## Enterprise Usage APIs to migrate off

The table below summarizes the different APIs that you may be using today to ingest cost details data. If you're using one of the APIs below, you'll need to migrate to one of the new solutions outlined above. All APIs below are behind the <https://consumption.azure.com> endpoint.

Endpoint	API Comments
/v3/enrollments/{enrollmentNumber}/usagedetails/download?billingPeriod={billingPeriod}	- API method: GET - Synchronous (non polling) - Data format: CSV
/v3/enrollments/{enrollmentNumber}/usagedetails/download?startTime=2017-01-01&endTime=2017-01-10	- API method: GET - Synchronous (non polling) - Data format: CSV
/v3/enrollments/{enrollmentNumber}/usagedetails	- API method: GET - Synchronous (non polling) - Data format: JSON

Endpoint	API Comments
/v3/enrollments/{enrollmentNumber}/billingPeriods/{billingPeriod}/usagedetails	<ul style="list-style-type: none"> <li>- API method: GET</li> <li>-</li> <li>Synchronous (non polling)</li> <li>- Data format: JSON</li> </ul>
/v3/enrollments/{enrollmentNumber}/usagedetailsbycustomdate?startTime=2017-01-01&endTime=2017-01-10	<ul style="list-style-type: none"> <li>- API method: GET</li> <li>-</li> <li>Synchronous (non polling)</li> <li>- Data format: JSON</li> </ul>
/v3/enrollments/{enrollmentNumber}/usagedetails/submit?billingPeriod={billingPeriod}	<ul style="list-style-type: none"> <li>- API method: POST</li> <li>-</li> <li>Asynchronous (polling based)</li> <li>- Data format: CSV</li> </ul>
/v3/enrollments/{enrollmentNumber}/usagedetails/submit?startTime=2017-04-01&endTime=2017-04-10	<ul style="list-style-type: none"> <li>- API method: POST</li> <li>-</li> <li>Asynchronous (polling based)</li> <li>- Data format: CSV</li> </ul>

## Enterprise Marketplace Store Charge APIs to migrate off

In addition to the usage details APIs outlined above, you'll need to migrate off the [Enterprise Marketplace Store Charge APIs](#). All Azure and Marketplace charges have been merged into a single file that is available through the new solutions. You can identify which charges are *Azure* versus *Marketplace* charges by using the `PublisherType` field that is available in the new dataset. The table below outlines the applicable APIs. All of the following APIs are behind the <https://consumption.azure.com> endpoint.

Endpoint	API Comments
/v3/enrollments/{enrollmentNumber}/marketplacecharges	- API method: GET  - Synchronous (non polling) - Data format: JSON
/v3/enrollments/{enrollmentNumber}/billingPeriods/{billingPeriod}/marketplacecharges	- API method: GET  - Synchronous (non polling) - Data format: JSON
/v3/enrollments/{enrollmentNumber}/marketplacechargesbycustomdate?startTime=2017-01-01&endTime=2017-01-10	- API method: GET  - Synchronous (non polling) - Data format: JSON

## Data field mapping

The table below provides a summary of the old fields available in the solutions you're currently using along with the field to use in the new solutions.

Old field	New field	Comments
serviceName	MeterCategory	
serviceTier	MeterSubCategory	
location	ResourceLocation	
chargesBilledSeparately	isAzureCreditEligible	The properties are opposites. If isAzureCreditEnabled is true, ChargesBilledSeparately would be false.
partNumber	PartNumber	

Old field	New field	Comments
resourceGuid	MeterId	Values vary. <code>resourceGuid</code> is a GUID value. <code>meterId</code> is a long number.
offerId	OfferId	
cost	CostInBillingCurrency	
accountId	AccountId	
resourceLocationId		Not available.
consumedServiceId	ConsumedService	<code>consumedServiceId</code> only provides a number value. <code>ConsumedService</code> provides the name of the service.
departmentId	InvoiceSectionId	
accountOwnerEmail	AccountOwnerId	
accountName	AccountName	
subscriptionId	SubscriptionId	
subscriptionGuid	SubscriptionId	
subscriptionName	SubscriptionName	
date	Date	
product	ProductName	
meterId	MeterId	
meterCategory	MeterCategory	
meterSubCategory	MeterSubCategory	
meterRegion	MeterRegion	
meterName	MeterName	
consumedQuantity	Quantity	
resourceRate	EffectivePrice	
resourceLocation	ResourceLocation	
consumedService	ConsumedService	
instanceId	ResourceId	

<b>Old field</b>	<b>New field</b>	<b>Comments</b>
serviceInfo1	ServiceInfo1	
serviceInfo2	ServiceInfo2	
additionalInfo	AdditionalInfo	
tags	Tags	
storeServiceIdentifier		Not available.
departmentName	InvoiceSectionName	
costCenter	CostCenter	
unitOfMeasure	UnitOfMeasure	
resourceGroup	ResourceGroup	
isRecurringCharge		Where applicable, use the Frequency and Term fields moving forward.
extendedCost	CostInBillingCurrency	
planName	PlanName	
publisherName	PublisherName	
orderNumber		Not available.
usageStartDate	Date	
usageEndDate	Date	

## Next steps

- Read the [Migrate from EA Reporting to Azure Resource Manager APIs overview](#) article.

# Migrate from EA Balance Summary API

Article • 07/17/2022

EA customers who were previously using the Enterprise Reporting consumption.azure.com API to get their balance summary need to migrate to a replacement Azure Resource Manager API. Instructions to do this are outlined below along with any contract differences between the old API and the new API.

## Assign permissions to an SPN to call the API

Before calling the API, you need to configure a Service Principal with the correct permission. You use the service principal to call the API. For more information, see [Assign permissions to ACM APIs](#).

## Call the Balance Summary API

Use the following request URIs when calling the new Balance Summary API. Your enrollment number should be used as the `billingAccountId`.

### Supported requests

#### Get for Enrollment

HTTP

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountId}/providers/Microsoft.Consumption/balances?api-version=2019-10-01
```

### Response body changes

Old response body:

JSON

```
{
 "id": "enrollments/100/billingperiods/201507/balancesummaries",
 "billingPeriodId": 201507,
 "currencyCode": "USD",
 "beginningBalance": 0,
 "endingBalance": 1.1,
 "newPurchases": 1,
```

```

"adjustments": 1.1,
"utilized": 1.1,
"serviceOverage": 1,
"chargesBilledSeparately": 1,
"totalOverage": 1,
"totalUsage": 1.1,
"azureMarketplaceServiceCharges": 1,
"newPurchasesDetails": [
{
 "name": "",
 "value": 1
},
],
"adjustmentDetails": [
{
 "name": "Promo Credit",
 "value": 1.1
},
{
 "name": "SIE Credit",
 "value": 1
}
]
}

```

New response body:

The same data is now available in the properties field of the new API response. There might be minor changes to the spelling on some of the field names.

JSON

```
{
 "id": "/providers/Microsoft.Billing/billingAccounts/123456/providers/Microsoft.Billing/billingPeriods/201702/providers/Microsoft.Consumption/balances/balanceId1",
 "name": "balanceId1",
 "type": "Microsoft.Consumption/balances",
 "properties": {
 "currency": "USD",
 "beginningBalance": 3396469.19,
 "endingBalance": 2922371.02,
 "newPurchases": 0,
 "adjustments": 0,
 "utilized": 474098.17,
 "serviceOverage": 0,
 "chargesBilledSeparately": 0,
 "totalOverage": 0,
 "totalUsage": 474098.17,
 "azureMarketplaceServiceCharges": 609.82,
 "billingFrequency": "Month",
 "priceHidden": false,
 "consumptionType": "Standard"
 }
}
```

```
"newPurchasesDetails": [
 {
 "name": "Promo Purchase",
 "value": 1
 }
],
"adjustmentDetails": [
 {
 "name": "Promo Credit",
 "value": 1.1
 },
 {
 "name": "SIE Credit",
 "value": 1
 }
]
```

## Next steps

- Read the [Migrate from EA Reporting to ARM APIs – Overview](#) article.

# Migrate from EA Price Sheet API

Article • 04/06/2023

EA customers who were previously using the Enterprise Reporting consumption.azure.com API to get their price sheet need to migrate to a replacement Azure Resource Manager API. Instructions to do this are outlined below along with any contract differences between the old API and the new API.

## Assign permissions to an SPN to call the API

Before calling the API, you need to configure a Service Principal with the correct permission. You use the service principal to call the API. For more information, see [Assign permissions to ACM APIs](#).

## Call the Price Sheet API

Use the following request URIs when calling the new Price Sheet API.

### Supported requests

You can call the API using the following scopes:

- Enrollment: `providers/Microsoft.Billing/billingAccounts/{billingAccountId}`
- Subscription: `subscriptions/{subscriptionId}`

#### [Get for current Billing Period](#)

HTTP

```
https://management.azure.com/{scope}/providers/Microsoft.Consumption/pricesheets/default?api-version=2019-10-01
```

#### [Get for specified Billing Period](#)

HTTP

```
https://management.azure.com/{scope}/billingPeriods/{billingPeriodName}/providers/Microsoft.Consumption/pricesheets/default?api-version=2019-10-01
```

## Response body changes

Old response:

JSON

```
[
 {
 "id":
 "enrollments/57354989/billingperiods/201601/products/343/pricesheets",
 "billingPeriodId": "201704",
 "meterId": "dc210ecb-97e8-4522-8134-2385494233c0",
 "meterName": "A1 VM",
 "unitOfMeasure": "100 Hours",
 "includedQuantity": 0,
 "partNumber": "N7H-00015",
 "unitPrice": 0.00,
 "currencyCode": "USD"
 },
 {
 "id":
 "enrollments/57354989/billingperiods/201601/products/2884/pricesheets",
 "billingPeriodId": "201404",
 "meterId": "dc210ecb-97e8-4522-8134-5385494233c0",
 "meterName": "Locally Redundant Storage Premium Storage -
Snapshots - AU East",
 "unitOfMeasure": "100 GB",
 "includedQuantity": 0,
 "partNumber": "N9H-00402",
 "unitPrice": 0.00,
 "currencyCode": "USD"
 },
 ...
]
```

New response:

Old data is now in the `pricesheets` field of the new API response. Meter details information is also provided.

JSON

```
{
 "id": "/subscriptions/00000000-0000-0000-0000-
000000000000/providers/Microsoft.Billing/billingPeriods/201702/providers/Mic
rosoft.Consumption/pricesheets/default",
 "name": "default",
 "type": "Microsoft.Consumption/pricesheets",
 "properties": {
 "nextLink": "https://management.azure.com/subscriptions/00000000-0000-
0000-0000-000000000000/providers/microsoft.consumption/pricesheets/default?
api-version=2018-01-
31&$skiptoken=AQAAAA%3D%3D&$expand=properties/pricesheets/meterDetails",
 "pricesheets": [
]
 }
}
```

```
{
 "billingPeriodId": "/subscriptions/00000000-0000-0000-0000-
000000000000/providers/Microsoft.Billing/billingPeriods/201702",
 "meterId": "00000000-0000-0000-0000-000000000000",
 "unitOfMeasure": "100 Hours",
 "includedQuantity": 100,
 "partNumber": "XX-11110",
 "unitPrice": 0.00000,
 "currencyCode": "EUR",
 "offerId": "OfferId 1",
 "meterDetails": {
 "meterName": "Data Transfer Out (GB)",
 "meterCategory": "Networking",
 "unit": "GB",
 "meterLocation": "Zone 2",
 "totalIncludedQuantity": 0,
 "pretaxStandardRate": 0.000
 }
}
}
]
}
```

## Next steps

- Read the [Migrate from EA Reporting to ARM APIs overview](#) article.

# Migrate from EA Reserved Instance Usage Details API

Article • 07/17/2022

EA customers who were previously using the Enterprise Reporting consumption.azure.com API to obtain reserved instance usage details need to migrate to a parity Azure Resource Manager API. Instructions to do this are outlined below along with any contract differences between the old API and the new API.

## Assign permissions to an SPN to call the API

Before calling the API, you need to configure a Service Principal with the correct permission. You use the service principal to call the API. For more information, see [Assign permissions to ACM APIs](#).

## Call the Reserved instance usage details API

Microsoft isn't updating the older synchronous-based Reservation Details APIs. We recommend at you move to the newer SPN-supported asynchronous API call pattern as a part of the migration. Asynchronous requests better handle large amounts of data and reduce timeout errors.

## Supported requests

Use the following request URIs when calling the new Asynchronous Reservation Details API. Your enrollment number should be used as the billingAccountId. You can call the API with the following scopes:

- Enrollment: `providers/Microsoft.Billing/billingAccounts/{billingAccountId}`

## Sample request to generate a reservation details report

HTTP

POST

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{bi
llingAccountId}/providers/Microsoft.CostManagement/generateReservationDetail
sReport?startDate={startDate}&endDate={endDate}&api-version=2019-11-01
```

## Sample request to poll report generation status

HTTP

GET

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountId}/providers/Microsoft.CostManagement/reservationDetailsOperationResults/{operationId}?api-version=2019-11-01
```

## Sample poll response

JSON

```
{
 "status": "Completed",
 "properties": {
 "reportUrl":
 "https://storage.blob.core.windows.net/details/20200911/00000000-0000-0000-0000-000000000000?sv=2016-05-31&sr=b&sig=jep8HT2aphfUkyERRZa5LRfd9RPzjXbzB%2F9TNiQ",
 "validUntil": "2020-09-12T02:56:55.5021869Z"
 }
}
```

## Response body changes

The response of the older synchronous based Reservation Details API is below.

Old response:

JSON

```
{
 "reservationOrderId": "00000000-0000-0000-0000-000000000000",
 "reservationId": "00000000-0000-0000-0000-000000000000",
 "usageDate": "2018-02-01T00:00:00",
 "skuName": "Standard_F2s",
 "instanceId": "/subscriptions/00000000-0000-0000-0000-000000000000/resourcegroups/resourcegroup1/providers/microsoft.compute/virtualmachines/VM1",
 "totalReservedQuantity": 18.00000000000000,
 "reservedHours": 432.0000000000000,
 "usedHours": 400.0000000000000
}
```

New response:

The new API creates a CSV file for you. See the following file fields.

Old property	New property	Notes
	InstanceFlexibilityGroup	New property for instance flexibility.
	InstanceFlexibilityRatio	New property for instance flexibility.
instanceId	InstanceName	
	Kind	It's a new property. Value is <code>None</code> , <code>Reservation</code> , or <code>IncludedQuantity</code> .
reservationId	ReservationId	
reservationOrderId	ReservationOrderId	
reservedHours	ReservedHours	
skuName	SkuName	
totalReservedQuantity	TotalReservedQuantity	
usageDate	UsageDate	
usedHours	UsedHours	

## Next steps

- Read the [Migrate from EA Reporting to ARM APIs overview](#) article.

# Migrate from EA Reserved Instance Usage Summary API

Article • 07/17/2022

EA customers who were previously using the Enterprise Reporting consumption.azure.com API to obtain reserved instance usage summaries need to migrate to a parity Azure Resource Manager API. Instructions to do this are outlined below along with any contract differences between the old API and the new API.

## Assign permissions to an SPN to call the API

Before calling the API, you need to configure a Service Principal with the correct permission. You use the service principal to call the API. For more information, see [Assign permissions to ACM APIs](#).

## Call the Reserved Instance Usage Summary API

Use the following request URIs to call the new Reservation Summaries API.

### Supported requests

Call the API with the following scopes:

- Enrollment: `providers/Microsoft.Billing/billingAccounts/{billingAccountId}`

#### [Get Reservation Summary Daily](#)

HTTP

```
https://management.azure.com/{scope}/Microsoft.Consumption/reservationSummaries?grain=daily&$filter=properties/usageDate ge 2017-10-01 AND properties/usageDate le 2017-11-20&api-version=2019-10-01
```

#### [Get Reservation Summary Monthly](#)

HTTP

```
https://management.azure.com/{scope}/Microsoft.Consumption/reservationSummaries?grain=daily&$filter=properties/usageDate ge 2017-10-01 AND properties/usageDate le 2017-11-20&api-version=2019-10-01
```

## Response body changes

Old response:

JSON

```
[
 {
 "reservationOrderId": "00000000-0000-0000-0000-000000000000",
 "reservationId": "00000000-0000-0000-0000-000000000000",
 "skuName": "Standard_F1s",
 "reservedHours": 24,
 "usageDate": "2018-05-01T00:00:00",
 "usedHours": 23,
 "minUtilizationPercentage": 0,
 "avgUtilizationPercentage": 95.83,
 "maxUtilizationPercentage": 100
 }
]
```

New response:

JSON

```
{
 "value": [
 {
 "id":
 "/providers/Microsoft.Billing/billingAccounts/12345/providers/Microsoft.Consumption/reservationSummaries/reservationSummaries_Id1",
 "name": "reservationSummaries_Id1",
 "type": "Microsoft.Consumption/reservationSummaries",
 "tags": null,
 "properties": {
 "reservationOrderId": "00000000-0000-0000-0000-000000000000",
 "reservationId": "00000000-0000-0000-0000-000000000000",
 "skuName": "Standard_B1s",
 "reservedHours": 720,
 "usageDate": "2018-09-01T00:00:00-07:00",
 "usedHours": 0,
 "minUtilizationPercentage": 0,
 "avgUtilizationPercentage": 0,
 "maxUtilizationPercentage": 0
 }
 }
]
 }
```

## Next steps

- Read the [Migrate from EA Reporting to ARM APIs overview](#) article.

# Migrate from EA Reserved Instance Recommendations API

Article • 07/17/2022

EA customers who were previously using the Enterprise Reporting consumption.azure.com API to obtain reserved instance recommendations need to migrate to a parity Azure Resource Manager API. Instructions to do this are outlined below along with any contract differences between the old API and the new API.

## Assign permissions to an SPN to call the API

Before calling the API, you need to configure a Service Principal with the correct permission. You use the service principal to call the API. For more information, see [Assign permissions to ACM APIs](#).

## Call the reserved instance recommendations API

Use the following request URIs to call the new Reservation Recommendations API.

### Supported requests

Call the API with the following scopes:

- Enrollment: `providers/Microsoft.Billing/billingAccounts/{billingAccountId}`
- Subscription: `subscriptions/{subscriptionId}`
- Resource Groups:  
`subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}`

#### [Get Recommendations](#)

Both the shared and the single scope recommendations are available through this API. You can also filter on the scope as an optional API parameter.

HTTP

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/123456/providers/Microsoft.Consumption/reservationRecommendations?api-version=2019-10-01
```

## Response body changes

Recommendations for Shared and Single scopes are combined into one API.

Old response:

JSON

```
[{
 "subscriptionId": "1111111-1111-1111-1111-111111111111",
 "lookBackPeriod": "Last7Days",
 "meterId": "2e3c2132-1398-43d2-ad45-1d77f6574933",
 "skuName": "Standard_DS1_v2",
 "term": "P1Y",
 "region": "westus",
 "costWithNoRI": 186.27634908960002,
 "recommendedQuantity": 9,
 "totalCostWithRI": 143.12931642978083,
 "netSavings": 43.147032659819189,
 "firstUsageDate": "2018-02-19T00:00:00"
}]
```

New response:

JSON

```
{
 "value": [
 {
 "id": "billingAccount/123456/providers/Microsoft.Consumption/reservationRecommendations/00000000-0000-0000-0000-000000000000",
 "name": "00000000-0000-0000-0000-000000000000",
 "type": "Microsoft.Consumption/reservationRecommendations",
 "location": "westus",
 "sku": "Standard_DS1_v2",
 "kind": "legacy",
 "properties": {
 "meterId": "00000000-0000-0000-0000-000000000000",
 "term": "P1Y",
 "costWithNoReservedInstances": 12.0785105,
 "recommendedQuantity": 1,
 "totalCostWithReservedInstances": 11.4899644807748,
 "netSavings": 0.588546019225182,
 "firstUsageDate": "2019-07-07T00:00:00-07:00",
 "scope": "Shared",
 "lookBackPeriod": "Last7Days",
 "instanceFlexibilityRatio": 1,
 "instanceFlexibilityGroup": "DSv2 Series",
 "normalizedSize": "Standard_DS1_v2",
 "recommendedQuantityNormalized": 1,
 }
 }
]
}
```

```
"skuProperties": [
 {
 "name": "Cores",
 "value": "1"
 },
 {
 "name": "Ram",
 "value": "1"
 }
]
},
]
```

## Next steps

- Read the [Migrate from EA Reporting to ARM APIs overview](#) article.

# Migrate from EA Reserved Instance Charges API

Article • 07/17/2022

EA customers who were previously using the Enterprise Reporting consumption.azure.com API to obtain reserved instance charges need to migrate to a parity Azure Resource Manager API. Instructions to do this are outlined below along with any contract differences between the old API and the new API.

## Assign permissions to an SPN to call the API

Before calling the API, you need to configure a Service Principal with the correct permission. You use the service principal to call the API. For more information, see [Assign permissions to ACM APIs](#).

## Call the Reserved Instance Charges API

Use the following request URIs to call the new Reserved Instance Charges API.

### Supported requests

#### Get Reservation Charges by Date Range

HTTP

```
https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountId}/providers/Microsoft.Consumption/reservationTransactions?&filter=properties/eventDate+ge+2020-05-20+AND+properties/eventDate+le+2020-05-30&api-version=2019-10-01
```

### Response body changes

Old response:

JSON

```
[
 {
 "purchasingEnrollment": "string",
 "armSkuName": "Standard_F1s",
 "term": "P1Y",
```

```

 "region": "eastus",
 "PurchasingSubscriptionGuid": "00000000-0000-0000-0000-
000000000000",
 "PurchasingSubscriptionName": "string",
 "accountName": "string",
 "accountOwnerEmail": "string",
 "departmentName": "string",
 "costCenter": "",
 "currentEnrollment": "string",
 "eventDate": "string",
 "reservationOrderId": "00000000-0000-0000-0000-000000000000",
 "description": "Standard_F1s eastus 1 Year",
 "eventType": "Purchase",
 "quantity": int,
 "amount": double,
 "currency": "string",
 "reservationOrderName": "string"
 }
]

```

New response:

JSON

```
{
 "value": [
 {
 "id": "/billingAccounts/123456/providers/Microsoft.Consumption/reservationtransactions/20190909091919",
 "name": "201909091919",
 "type": "Microsoft.Consumption/reservationTransactions",
 "tags": {},
 "properties": {
 "eventDate": "2019-09-09T19:19:04Z",
 "reservationOrderId": "00000000-0000-0000-0000-000000000000",
 "description": "Standard_DS1_v2 westus 1 Year",
 "eventType": "Cancel",
 "quantity": 1,
 "amount": -21,
 "currency": "USD",
 "reservationOrderName": "Transaction-DS1_v2",
 "purchasingEnrollment": "123456",
 "armSkuName": "Standard_DS1_v2",
 "term": "P1Y",
 "region": "westus",
 "purchasingSubscriptionGuid": "11111111-1111-1111-1111-111111111111",
 "purchasingSubscriptionName": "Infrastructure Subscription",
 "accountName": "Microsoft Infrastructure",
 "accountOwnerEmail": "admin@microsoft.com",
 "departmentName": "Unassigned",
 "costCenter": "",
 "currentEnrollment": "123456",
 "currentEnrollment": "123456"
 }
 }
]
}
```

```
 "billingFrequency": "recurring"
 },
},
]
}
```

## Next steps

- Read the [Migrate from EA Reporting to ARM APIs overview](#) article.

# Migrate from Enterprise Agreement to Microsoft Customer Agreement APIs

Article • 09/21/2022

This article helps you understand the data structure, API, and other system integration differences between Enterprise Agreement (EA) and Microsoft Customer Agreement (MCA) accounts. Cost Management supports APIs for both account types. Review the [Setup billing account for Microsoft Customer Agreement](#) article before continuing.

Organizations with an existing EA account should review this article when they set up an MCA account. Previously, renewing an EA account required some minimal work to move from an old enrollment to a new one. However, migrating to an MCA account requires extra effort. Extra effort is because of changes in the underlying billing subsystem, which affect all cost-related APIs and service offerings.

## MCA APIs and integration

MCA APIs and new integration allow you to:

- Have complete API availability through native Azure APIs.
- Configure multiple invoices in a single billing account.
- Access a combined API with Azure service usage, third-party Marketplace usage, and Marketplace purchases.
- View costs across billing profiles (the same as enrollments) using Cost Management.
- Access new APIs to show costs, get notified when costs exceed predefined thresholds, and export raw data automatically.

## Migration checklist

The following items help you transition to MCA APIs.

- Familiarize yourself with the new [Microsoft Customer Agreement billing account](#).
- Determine which APIs you use and see which ones are replaced in the following section.
- Familiarize yourself with [Azure Resource Manager REST APIs](#).
- If you're not already using Azure Resource Manager APIs, [register your client app with Azure AD](#).
- Grant the application that was created during Azure AD app registration read access to the billing account using Access control (IAM).
- Update any programming code to [use Azure AD authentication](#).
- Update any programming code to replace EA API calls with MCA API calls.
- Update error handling to use new error codes.
- Review other integration offerings like Power BI for other needed action.

## EA APIs replaced with MCA APIs

EA APIs use an API key for authentication and authorization. MCA APIs use Azure AD authentication.

Purpose	EA API	MCA API
Balance and credits	<a href="#">/balancesummary</a>	Microsoft.Billing/billingAccounts/billingProfiles/availableBalanceuss
Usage (JSON)	<a href="#">/usagedetails/usagedetailsbycustomdate</a>	<a href="#">Choose a cost details solution</a>
Usage (CSV)	<a href="#">/usagedetails/download/usagedetails/submit</a>	<a href="#">Choose a cost details solution</a>
Marketplace Usage (CSV)	<a href="#">/marketplacecharges/marketplacechargesbycustomdate</a>	<a href="#">Choose a cost details solution</a>
Billing periods	<a href="#">/billingperiods</a>	Microsoft.Billing/billingAccounts/billingProfiles/invoices
Price sheet	<a href="#">/pricesheet</a>	Microsoft.Billing/billingAccounts/billingProfiles/pricesheet/default/c Microsoft.Billing/billingAccounts/.../billingProfiles/.../invoices/... /pri format=json csv Microsoft.Billing/billingAccounts/..../billingProfiles/..../providers/Micro
Reservation purchases	<a href="#">/reservationcharges</a>	Microsoft.Billing/billingAccounts/billingProfiles/transactions
Reservation recommendations	<a href="#">/SharedReservationRecommendations/SingleReservationRecommendations</a>	<a href="#">Microsoft.Consumption/reservationRecommendations</a>

Purpose	EA API	MCA API
Reservation usage	/reservationdetails/reservationsummaries	Microsoft.Consumption/reservationDetailsMicrosoft.Consumption/r

<sup>1</sup> Azure service and third-party Marketplace usage are available with the [Usage Details API](#).

The following APIs are available to MCA billing accounts:

Purpose	Microsoft Customer Agreement (MCA) API
Billing accounts <sup>2</sup>	Microsoft.Billing/billingAccounts
Billing profiles <sup>2</sup>	Microsoft.Billing/billingAccounts/billingProfiles
Invoice sections <sup>2</sup>	Microsoft.Billing/billingAccounts/invoiceSections
Invoices	Microsoft.Billing/billingAccounts/billingProfiles/invoices
Billing subscriptions	{scope}/billingSubscriptions

<sup>2</sup> APIs return lists of objects, which are scopes, where Cost Management experiences in the Azure portal and APIs operate. For more information about Cost Management scopes, see [Understand and work with scopes](#).

If you use any existing EA APIs, you need to update them to support MCA billing accounts. The following table shows other integration changes:

Purpose	Old offering	New offering
Power BI	<a href="#">Microsoft Consumption Insights</a> content pack and connector	<a href="#">Azure Consumption Insights connector</a>

## APIs to get balance and credits

The [Get Balance Summary](#) API gives you a monthly summary of:

- Balances
- New purchases
- Azure Marketplace service charges
- Adjustments
- Service overage charges

All Consumption APIs are replaced by native Azure APIs that use Azure AD for authentication and authorization. For more information about calling Azure REST APIs, see [Getting started with REST](#).

The Get Balance Summary API is replaced by the Microsoft.Billing/billingAccounts/billingProfiles/availableBalance API.

To get available balances with the Available Balance API:

Method	Request URI
GET	<code>https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountId}/billingProfiles/{billingProfileId}/availableBalances?api-version=2018-11-01-preview</code>

## APIs to get cost and usage

Get a daily breakdown of costs from Azure service usage, third-party Marketplace usage, and other Marketplace purchases with the following APIs. The following separate APIs were merged for Azure services and third-party Marketplace usage. The old APIs are replaced by either [Exports](#) or the [Cost Details API](#). To choose the solution that's right for you, see [Choose a cost details solution](#). Both solutions provide the same Cost Details file and have marketplace purchases in the data, which were previously only shown in the balance summary to date.

- [Get usage detail/download](#)
- [Get usage detail/submit](#)
- [Get usage detail/usagedetails](#)
- [Get usage detail/usagedetailsbycustomdate](#)
- [Get marketplace store charge/marketplacecharges](#)
- [Get marketplace store charge/marketplacechargesbycustomdate](#)

Exports and the Cost Details API, as with all Cost Management APIs, are available at multiple scopes. For invoiced costs, as you would traditionally receive at an enrollment level, use the billing profile scope. For more information about Cost Management scopes, see [Understand and work with scopes](#).

Type	ID format
Billing account	/Microsoft.Billing/billingAccounts/{billingAccountId}
Billing profile	/Microsoft.Billing/billingAccounts/{billingAccountId}/billingProfiles/{billingProfileId}
Subscription	/subscriptions/{subscriptionId}
Resource group	/subscriptions/{subscriptionId}/resourceGroups/{resourceGroupName}

Some property names have changed in the new Cost Details dataset available through Exports and Cost Details API. The following table shows corresponding properties.

Old property	New property	Notes
Accountid	N/A	The subscription creator isn't tracked. Use invoiceSectionId (same as departmentId).
AccountNameAccountOwnerId and AccountOwnerEmail	N/A	The subscription creator isn't tracked. Use invoiceSectionName (same as departmentName).
AdditionallInfo	additionalInfo	
ChargesBilledSeparately	isAzureCreditEligible	The properties are opposites. If isAzureCreditEnabled is true, ChargesBilledSeparately would be false.
ConsumedQuantity	quantity	
ConsumedService	consumedService	Exact string values might differ.
ConsumedServiceId	None	
CostCenter	costCenter	
Date and usageStartDate	date	
Day	None	Parses day from date.
DepartmentId	invoiceSectionId	Exact values differ.
DepartmentName	invoiceSectionName	Exact string values might differ. Configure invoice sections to match departments, if needed.
ExtendedCost and Cost	costInBillingCurrency	
Instanceld	resourceId	
Is Recurring Charge	None	
Location	location	
MeterCategory	meterCategory	Exact string values might differ.
MeterId	meterId	Exact string values differ.
MeterName	meterName	Exact string values might differ.
MeterRegion	meterRegion	Exact string values might differ.
MeterSubCategory	meterSubCategory	Exact string values might differ.
Month	None	Parses month from date.
Offer Name	None	Use publisherName and productOrderName.
OfferID	None	
Order Number	None	
PartNumber	None	Use meterId and productOrderName to uniquely identify prices.
Plan Name	productOrderName	
Product	Product	

Old property	New property	Notes
ProductId	productId	Exact string values differ.
Publisher Name	publisherName	
ResourceGroup	resourceGroupName	
ResourceGuid	meterId	Exact string values differ.
ResourceLocation	resourceLocation	
ResourceLocationId	None	
ResourceRate	effectivePrice	
ServiceAdministratorId	N/A	
ServiceInfo1	serviceInfo1	
ServiceInfo2	serviceInfo2	
ServiceName	meterCategory	Exact string values might differ.
ServiceTier	meterSubCategory	Exact string values might differ.
StoreServiceIdentifier	N/A	
SubscriptionGuid	subscriptionId	
SubscriptionId	subscriptionId	
SubscriptionName	subscriptionName	
Tags	tags	The tags property applies to the root object, not to the properties nested property.
UnitOfMeasure	unitOfMeasure	Exact string values differ.
usageEndDate	date	
Year	None	Parses year from date.
(new)	billingCurrency	Currency used for the charge.
(new)	billingProfileId	Unique ID for the billing profile (same as enrollment).
(new)	billingProfileName	Name of the billing profile (same as enrollment).
(new)	chargeType	Use to differentiate Azure service usage, Marketplace usage, and purchases.
(new)	invoiceId	Unique ID for the invoice. Empty for the current, open month.
(new)	publisherType	Type of publisher for purchases. Empty for usage.
(new)	serviceFamily	Type of purchase. Empty for usage.
(new)	servicePeriodEndDate	End date for the purchased service.
(new)	servicePeriodStartDate	Start date for the purchased service.

## Billing Periods API replaced by Invoices API

MCA billing accounts don't use billing periods. Instead, they use invoices to scope costs to specific billing periods. The [Billing Periods API](#) is replaced by the Invoices API. All Consumption APIs are replaced by native Azure APIs that use Azure AD for authentication and authorization. For more information about calling Azure REST APIs, see [Getting started with REST](#).

To get invoices with the Invoices API:

Method	Request URI
GET	<code>https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountId}/billingProfiles/{billingProfileId}/invoices?</code> <code>api-version=2018-11-01-preview</code>

## Price Sheet APIs

This section discusses existing Price Sheet APIs and provides recommendations to move to the Price Sheet API for Microsoft Customer Agreements. It also discusses the Price Sheet API for Microsoft Customer Agreements and explains fields in the price sheets. The [Enterprise Get price sheet](#) and [Enterprise Get billing periods](#) APIs are replaced by the Price Sheet API for Microsoft Customer Agreements (`Microsoft.Billing/billingAccounts/billingProfiles/pricesheet`). The new API supports both JSON and CSV formats, in asynchronous REST formats. All Consumption APIs are replaced by native Azure APIs that use Azure AD for authentication and authorization. For more information about calling Azure REST APIs, see [Getting started with REST](#).

## Billing Enterprise APIs

You used Billing Enterprise APIs with Enterprise enrollments to get price and billing period information. Authentication and authorization used Azure Active Directory web tokens.

To get applicable prices for the specified Enterprise Enrollment with the Price Sheet and Billing Period APIs:

Method	Request URI
GET	<code>https://consumption.azure.com/v2/enrollments/{enrollmentNumber}/pricesheet</code>
GET	<code>https://consumption.azure.com/v2/enrollments/{enrollmentNumber}/billingPeriods/{billingPeriod}/pricesheet</code>

## Price Sheet API for Microsoft Customer Agreements

Use the Price Sheet API for Microsoft Customer Agreements to view prices for all Azure Consumption and Marketplace consumption services. The prices shown for the billing profile apply to all subscriptions that belong to the billing profile.

Use the Price Sheet API to view all Azure Consumption services Price Sheet data in CSV format:

Method	Request URI
POST	<code>https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountId}/billingProfiles/{billingProfileId}/pricesheet/default/?api-version=2018-11-01-preview&amp;startDate=2019-01-01&amp;endDate=2019-01-31&amp;format=csv</code>

Use the Price Sheet API to view all Azure Consumption services Price Sheet data in JSON format:

Method	Request URI
POST	<code>https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountId}/billingProfiles/{billingProfileId}/pricesheet/default/?api-version=2018-11-01-preview&amp;startDate=2019-01-01&amp;endDate=2019-01-31&amp;format=json</code>

Using the API returns the price sheet for the entire account. However, you can also get a condensed version of the price sheet in PDF format. The summary includes Azure Consumption and Marketplace consumption services that are billed for a specific invoice. The invoice is identified by the `{invoiceId}`, which is the same as the **Invoice Number** shown in the Invoice Summary PDF files. Here's an example.

Microsoft		Invoice									
Microsoft Corporation One Microsoft Way Redmond WA 98052 United States FEIN: 91-1144442		<b>Invoice Summary</b> <table> <tr> <td>Billing Profile</td> <td>PayByCheck_CA</td> </tr> <tr> <td>Invoice Number</td> <td>T000006608</td> </tr> <tr> <td>Invoice Date</td> <td>01/31/2019</td> </tr> <tr> <td>Payment Terms</td> <td>Net 30 days</td> </tr> </table>		Billing Profile	PayByCheck_CA	Invoice Number	T000006608	Invoice Date	01/31/2019	Payment Terms	Net 30 days
Billing Profile	PayByCheck_CA										
Invoice Number	T000006608										
Invoice Date	01/31/2019										
Payment Terms	Net 30 days										
Sold To	Bill To	<b>Total Amount</b> <b>USD 676.00</b> <b>Due on 03/02/2019</b>									
Contoso_Test_California 1020 Enterprise Way Sunnyvale ca 94089 US	Contoso_Test_California 1020 Enterprise Way Sunnyvale ca 94089 US	<small>Questions on your bill? Visit <a href="https://aka.ms/invoice-billing">https://aka.ms/invoice-billing</a></small>									

To view invoice information with the Price Sheet API in CSV format:

Method	Request URI
POST	<code>https://management.azure.com/providers/Microsoft.Billing/billingAccounts/2909cffc-b0a2-5de1-bb7b-5d3383764184/billingProfiles/2dcffe0c-ee92-4265-8647-515b8fe7dc78/invoices/{invoiceId}/pricesheet/default/download?api-version=2018-11-01-preview&amp;format=csv</code>

To view invoice information with the Price Sheet API in JSON Format:

Method	Request URI
POST	<code>https://management.azure.com/providers/Microsoft.Billing/billingAccounts/2909cffc-b0a2-5de1-bb7b-5d3383764184/billingProfiles/2dcffe0c-ee92-4265-8647-515b8fe7dc78/invoices/{invoiceId}/pricesheet/default/download?api-version=2018-11-01-preview&amp;format=json</code>

You can also see estimated prices for any Azure Consumption or Marketplace consumption service in the current open billing cycle or service period.

To view estimated prices for consumption services with the Price Sheet API in CSV format:

Method	Request URI
POST	<code>https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountId}/billingProfiles/{billingProfileId}/pricesheet/default/download?api-version=2018-11-01-preview&amp;format=csv</code>

To view estimated prices for consumption services with the Price Sheet API in JSON format:

Method	Request URI
POST	<code>https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountId}/billingProfiles/{billingProfileId}/pricesheet/default/download?api-version=2018-11-01-preview&amp;format=json</code>

The Microsoft Customer Agreement Price Sheet APIs are *asynchronous REST APIs*. The responses for the APIs changed from the older synchronous APIs. The body of the API response also changed.

## Old response body

Here's an example of the synchronous REST API response:

[
{
"id": "enrollments/573549891/billingperiods/2016011/products/343/pricesheets",
"billingPeriodId": "201704",
"meterId": "dc210ecb-97e8-4522-8134-2385494233c0",
"meterName": "A1 VM",
"unitOfMeasure": "100 Hours",
"includedQuantity": 0,
"partNumber": "N7H-00015",
"unitPrice": 0.00,
"currencyCode": "USD"
},
{
]

## New response body

The APIs support the [Azure REST asynchronous](#) format. Call the API using GET and you receive the following response:

No Response Body
HTTP Status 202 Accepted

The following headers are sent with the location of the output:

Location: <code>https://management.azure.com/providers/Microsoft.Consumption/operationresults/{operationId}?</code>
sessiontoken=XZDFSnvdkbkdsb==
Azure-AsyncOperation: <code>https://management.azure.com/providers/Microsoft.Consumption/operationStatus/{operationId}?</code>
sessiontoken=XZDFSnvdkbkdsb==
Retry-After: 10

```
OData-EntityId: {operationId}
```

Make another GET call to the location. The response to the GET call is the same until the operation reaches a completion or failure state. When completed, the response to the GET call location returns the download URL as if the operation was executed at the same time. Here's an example:

```
HTTP Status 200

{
 "id": "providers/Microsoft.Consumption/operationresults/{operationId}",
 "name": {operationId},
 "type": "Microsoft.Consumption/operationResults",
 "properties" : {
 "downloadUrl": {urltoblob},
 "validTill": "Date"
 }
}
```

The client can also make a GET call for the [Azure-AsyncOperation](#). The endpoint returns the status for the operation.

The following table shows fields in the older Enterprise Get price sheet API. It includes corresponding fields in the new price sheet for Microsoft Customer Agreements:

Old property	New property	Notes
billingPeriodId	<i>Not applicable</i>	Not applicable. For Microsoft Customer Agreements, the invoice and associated price sheet replaced the concept of billingPeriodId.
meterId	meterId	
unitOfMeasure	unitOfMeasure	Exact string values might differ.
includedQuantity	includedQuantity	Not applicable for services in Microsoft Customer Agreements.
partNumber	<i>Not applicable</i>	Instead, use a combination of productOrderName (same as offerID) and meterID.
unitPrice	unitPrice	Unit price is applicable for services consumed in Microsoft Customer Agreements.
currencyCode	pricingCurrency	Microsoft Customer Agreements have price representations in pricing currency and billing currency. The currencyCode corresponds to the pricingCurrency in Microsoft Customer Agreements.
offerID	productOrderName	Instead of OfferID, you can use productOrderName but isn't the same as OfferID. However, productOrderName and meter determine pricing in Microsoft Customer Agreements related to meterId and OfferID in legacy enrollments.

## Consumption Price Sheet API operations

For Enterprise Agreements, you used the Consumption Price Sheet API [Get](#) and [Get By Billing Period](#) operations for a scope by subscriptionId or a billing period. The API uses Azure Resource Management authentication.

To get the Price Sheet information for a scope with the Price Sheet API:

Method	Request URI
GET	<a href="https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.Consumption/pricesheets/default?api-version=2018-10-01">https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.Consumption/pricesheets/default?api-version=2018-10-01</a>

To get Price Sheet information by billing period with the Price Sheet API:

Method	Request URI
GET	<a href="https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.Billing/billingPeriods/{billingPeriodName}/providers/Microsoft.Consumption/pricesheets/default?api-version=2018-10-01">https://management.azure.com/subscriptions/{subscriptionId}/providers/Microsoft.Billing/billingPeriods/{billingPeriodName}/providers/Microsoft.Consumption/pricesheets/default?api-version=2018-10-01</a>

Instead of the above API endpoints, use the following ones for Microsoft Customer Agreements:

### Price Sheet API for Microsoft Customer Agreements (asynchronous REST API)

This API is for Microsoft Customer Agreements and it provides extra attributes.

## Price Sheet for a Billing Profile scope in a Billing Account

This API is the existing API. It was updated to provide the price sheet for a billing profile in a billing account.

## Price Sheet for a scope by billing account

Azure Resource Manager authentication is used when you get the Price Sheet at the enrollment scope in a billing account.

To get the Price Sheet at the enrollment account in a billing account:

Method	Request URI
GET	/providers/Microsoft.Billing/billingAccounts/65085863/providers/Microsoft.Consumption/pricesheets/download?api-version=2019-01-01

For a Microsoft Customer Agreement, use the information in the following section. It provides the field properties used for Microsoft Customer agreements.

## Price Sheet for a billing profile scope in a billing account

The updated Price Sheet by billing account API gets the Price Sheet in CSV format. To get the Price Sheet at the billing profile scope for an MCA:

Method	Request URI
GET	/providers/Microsoft.Billing/billingAccounts/28ae4b7f-41bb-581e-9fa4-8270c857aa5f/billingProfiles/ef37facb-cd6f-437a-9261-65df15b673f9/providers/Microsoft.Consumption/pricesheets/download?api-version=2019-01-01

At the EA's enrollment scope, the API response and properties are identical. The properties correspond to the same MCA properties.

The older properties for [Azure Resource Manager Price Sheet APIs](#) and the same new properties are in the following table.

Old Azure Resource Manager Price Sheet API Property	New Microsoft Customer Agreement Price Sheet API property	Description
Meter ID	<i>meterId</i>	Unique identifier for the meter. Same as meterID.
Meter name	<i>meterName</i>	Name of the meter. Meter represents the Azure service deployable resource.
Meter category	<i>service</i>	Name of the classification category for the meter. Same as the service in the Microsoft Customer Agreement Price Sheet. Exact string values differ.
Meter subcategory	<i>meterSubCategory</i>	Name of the meter subclassification category. Based on the classification of high-level feature set differentiation in the service. For example, Basic SQL DB vs Standard SQL DB.
Meter region	<i>meterRegion</i>	
Unit	<i>Not applicable</i>	Can be parsed from unitOfMeasure.
Unit of measure	<i>unitOfMeasure</i>	
Part number	<i>Not applicable</i>	Instead of part number, use productOrderName and MeterID to uniquely identify the price for a billing profile. Fields are listed on the MCA invoice instead of the part number in MCA invoices.
Unit price	<i>unitPrice</i>	Microsoft Customer Agreement unit price.
Currency code	<i>pricingCurrency</i>	Microsoft Customer Agreements represent prices in pricing currency and billing currency. Currency code is the same as the pricingCurrency in Microsoft Customer Agreements.
Included quantity	<i>includedQuantity</i>	Not applicable to services in Microsoft Customer Agreements. Show with values of zero.
Offer ID	<i>productOrderName</i>	Instead of OfferID, use productOrderName. Not the same as OfferID, however the productOrderName and meter determine pricing in Microsoft Customer Agreements. Related to meterId and OfferID in legacy enrollments.

The price for Microsoft Customer Agreements is defined differently than Enterprise agreements. The price for services in the Enterprise enrollment is unique for product, part number, meter, and offer. The part number isn't used in Microsoft Customer Agreements.

The Azure Consumption service price that's part of a Microsoft Customer Agreement is unique for productOrderName and meterID. They represent the service meter and the product plan.

To reconcile between the price sheet and the usage in the Usage Details API, you can use the productOrderName and meterID.

Users that have billing profile owner, contributor, reader, and invoice manager rights can download the price sheet.

The price sheet includes prices for services whose price is based on usage. The services include Azure consumption and Marketplace consumption. The latest price at the end of each service period is locked and applied to usage in a single service period. For Azure consumption services, the service period is usually a calendar month.

## Retired Price Sheet API fields

The following fields are either not available in Microsoft Customer Agreement Price Sheet APIs or have the same fields.

Retired field	Description
billingPeriodId	No applicable. Corresponds to InvoiceId for MCA.
offerID	Not applicable. Corresponds to productOrderName in MCA.
meterCategory	Not applicable. Corresponds to Service in MCA.
unit	Not applicable. Can be parsed from unitOfMeasure.
currencyCode	Same as the pricingCurrency in MCA.
meterLocation	Same as the meterRegion in MCA.
partNumber	Not applicable because part number isn't listed in MCA invoices. Instead of part number, use the meterId and productOrderName combination to uniquely identify prices.
totalIncludedQuantity	Not applicable.
pretaxStandardRate	Not applicable.

## Reservation Instance Charge API replaced

You can get billing transactions for reservation purchases with the [Reserved Instance Charge API](#). The new API includes all purchases, including third-party Marketplace offerings. All Consumption APIs are replaced by native Azure APIs that use Azure AD for authentication and authorization. For more information about calling Azure REST APIs, see [Getting started with REST](#). The Reserved Instance Charge API is replaced by the Transactions API.

To get reservation purchase transactions with the Transactions API:

Method	Request URI
GET	<code>https://management.azure.com/providers/Microsoft.Billing/billingAccounts/{billingAccountId}/billingProfiles/{billingProfileId}/transactions?api-version=2018-11-01-preview</code>

## Recommendations APIs replaced

Reserved Instance Purchase Recommendations APIs provide virtual machine usage over the last 7, 30, or 60 days. APIs also provide reservation purchase recommendations. They include:

- [Shared Reserved Instance Recommendation API](#)
- [Single Reserved Instance Recommendations API](#)

All Consumption APIs are replaced by native Azure APIs that use Azure AD for authentication and authorization. For more information about calling Azure REST APIs, see [Getting started with REST](#). The reservation recommendations APIs listed previously are replaced by the [Microsoft.Consumption/reservationRecommendations](#) API.

To get reservation recommendations with the Reservation Recommendations API:

Method	Request URI
GET	<code>https://management.azure.com/providers/Microsoft.Consumption/reservationRecommendations?api-version=2019-01-01</code>

## Reservation Usage APIs replaced

You can get reservation usage in an enrollment with the Reserved Instance Usage API. If there's more than one reserved instance in an enrollment, you can also get the usage of all the reserved instance purchases using this API.

They include:

- [Reserved Instance Usage Details](#)
- [Reserved Instance Usage Summary](#)

All Consumption APIs are replaced by native Azure APIs that use Azure AD for authentication and authorization. For more information about calling Azure REST APIs, see [Getting started with REST](#). The reservation recommendations APIs listed previously are replaced by the [Microsoft.Consumption/reservationDetails](#) and [Microsoft.Consumption/reservationSummaries](#) APIs.

To get reservation details with the Reservation Details API:

Method	Request URI
GET	<a href="https://management.azure.com/providers/Microsoft.Consumption/reservationDetails?api-version=2019-01-01">https://management.azure.com/providers/Microsoft.Consumption/reservationDetails?api-version=2019-01-01</a>

To get reservation summaries with the Reservation Summaries API:

Method	Request URI
GET	<a href="https://management.azure.com/providers/Microsoft.Consumption/reservationSummaries?api-version=2019-01-01">https://management.azure.com/providers/Microsoft.Consumption/reservationSummaries?api-version=2019-01-01</a>

## Power BI integration

You can also use Power BI for cost reporting. The [Cost Management connector](#) for Power BI Desktop can be used to create powerful, customized reports that help you better understand your Azure spend. The Cost Management connector currently supports customers with either a Microsoft Customer Agreement or an Enterprise Agreement (EA).

## Next steps

- Read the [Cost Management documentation](#) to learn how to monitor and control Azure spending. Or, if you want to optimize resource use with Cost Management.

# Migrate from Consumption Usage Details API

Article • 09/21/2022

This article discusses migration away from the [Consumption Usage Details API](#). The Consumption Usage Details API is deprecated. The date that the API will be turned off is still being determined. We recommend that you migrate away from the API as soon as possible.

## Migration destinations

Read the [Choose a cost details solution](#) article before you choose which solution is right for your workload. Generally, we recommend [Exports](#) if you have ongoing data ingestion needs and or a large monthly usage details dataset. For more information, see [Ingest usage details data](#).

If you have a smaller usage details dataset or a scenario that isn't met by Exports, consider using the [Cost Details](#) report instead. For more information, see [Get small cost datasets on demand](#).

### Note

The [Cost Details](#) report is only available for customers with an Enterprise Agreement or Microsoft Customer Agreement. If you have an MSDN, pay-as-you-go, or Visual Studio subscription, you can migrate to Exports or continue using the Consumption Usage Details API.

## Migration benefits

New solutions provide many benefits over the Consumption Usage Details API. Here's a summary:

- **Single dataset for all usage details** - Azure and Azure Marketplace usage details were merged into one dataset. It reduces the number of APIs that you need to call to get see all your charges.
- **Scalability** - The Marketplaces API is deprecated because it promotes a call pattern that isn't able to scale as your Azure usage increases. The usage details dataset can get extremely large as you deploy more resources into the cloud. The

Marketplaces API is a paginated synchronous API so it isn't optimized to effectively transfer large volumes of data over a network with high efficiency and reliability. Exports and the [Cost Details](#) API are asynchronous. They provide you with a CSV file that can be directly downloaded over the network.

- **API improvements** - Exports and the Cost Details API are the solutions that Azure supports moving forward. All new features are being integrated into them.
  - **Schema consistency** - The [Cost Details](#) report and [Exports](#) provide files with matching fields so you can move from one solution to the other, based on your scenario.
  - **Cost Allocation integration** - Enterprise Agreement and Microsoft Customer Agreement customers using Exports or the Cost Details API can view charges in relation to the cost allocation rules that they have configured. For more information about cost allocation, see [Allocate costs](#).

# Field Differences

The following table summarizes the field differences between the Consumption Usage Details API and Exports/Cost Details API. Exports and the Cost Details API provide a CSV file download instead of the paginated JSON response that's provided by the Consumption API.

## Enterprise Agreement field mapping

Enterprise Agreement customers who are using the Consumption Usage Details API have usage details records of the kind `Legacy`. A legacy usage details record is shown below. All Enterprise Agreement customers have records of this kind due to the underlying billing system that's used for them.

## JSON

```

 "tags": {

 "env": "newcrp",

 "dev": "tools"

 },

 "properties": {

 }

}

```

A full example legacy Usage Details record is shown at [Usage Details - List - REST API \(Azure Consumption\)](#)

The following table provides a mapping between the old and new fields. New properties are available in the CSV files produced by Exports and the Cost Details API. To learn more about the fields, see [Understand usage details fields](#).

**Bold** property names are unchanged.

Old Property	New Property
accountName	AccountName
<b>AccountOwnerId</b>	AccountOwnerId
additionallInfo	AdditionalInfo
<b>AvailabilityZone</b>	AvailabilityZone
billingAccountId	BillingAccountId
billingAccountName	BillingAccountName
billingCurrency	BillingCurrencyCode
billingPeriodEndDate	BillingPeriodEndDate
billingPeriodStartDate	BillingPeriodStartDate
billingProfileId	BillingProfileId
billingProfileName	BillingProfileName
chargeType	ChargeType

Old Property	New Property
consumedService	ConsumedService
cost	CostInBillingCurrency
costCenter	CostCenter
date	Date
effectivePrice	EffectivePrice
frequency	Frequency
invoiceSection	InvoiceSectionName
<b>InvoiceSectionId</b>	InvoiceSectionId
isAzureCreditEligible	IsAzureCreditEligible
meterCategory	MeterCategory
meterId	MeterId
meterName	MeterName
<b>MeterRegion</b>	MeterRegion
meterSubCategory	MeterSubCategory
offerId	OfferId
partNumber	PartNumber
<b>PayGPrice</b>	PayGPrice
<b>PlanName</b>	PlanName
<b>PricingModel</b>	PricingModel
product	ProductName
<b>ProductOrderId</b>	ProductOrderId
<b>ProductOrderName</b>	ProductOrderName
<b>PublisherName</b>	PublisherName
<b>PublisherType</b>	PublisherType
quantity	Quantity
<b>ReservationId</b>	ReservationId

Old Property	New Property
ReservationName	ReservationName
resourceGroup	ResourceGroup
resourceId	ResourceId
resourceLocation	ResourceLocation
resourceName	ResourceName
serviceFamily	ServiceFamily
ServiceInfo1	ServiceInfo1
ServiceInfo2	ServiceInfo2
subscriptionId	SubscriptionId
subscriptionName	SubscriptionName
Tags	Tags
Term	Term
unitOfMeasure	UnitOfMeasure
unitPrice	UnitPrice
CostAllocationRuleName	CostAllocationRuleName

## Microsoft Customer Agreement field mapping

Microsoft Customer Agreement customers that use the Consumption Usage Details API have usage details records of the kind `modern`. A modern usage details record is shown below. All Microsoft Customer Agreement customers have records of this kind due to the underlying billing system that is used for them.

JSON

```
{
 "value": [
 {
 "id": "{id}",
 "name": "{name}",
 "type": "Usage"
 }
]
}
```

```

 "type": "Microsoft.Consumption/usageDetails",
 "kind": "modern",
 "tags": {
 "env": "newcrp",
 "dev": "tools"
 },
 "properties": {

 }
}

```

An full example legacy Usage Details record is shown at [Usage Details - List - REST API \(Azure Consumption\)](#)

A mapping between the old and new fields are shown in the following table. New properties are available in the CSV files produced by Exports and the Cost Details API. Fields that need a mapping due to differences across the solutions are shown in **bold text**.

For more information, see [Understand usage details fields](#).

Old property	New property
invoiceld	invoiceld
previousInvoiceld	previousInvoiceld
billingAccountId	billingAccountId
billingAccountName	billingAccountName
billingProfileId	billingProfileId
billingProfileName	billingProfileName
invoiceSectionId	invoiceSectionId
invoiceSectionName	invoiceSectionName
partnerTenantId	partnerTenantId

<b>Old property</b>	<b>New property</b>
partnerName	partnerName
resellerName	resellerName
resellerMpnId	resellerMpnId
customerTenantId	customerTenantId
customerName	customerName
costCenter	costCenter
billingPeriodEndDate	billingPeriodEndDate
billingPeriodStartDate	billingPeriodStartDate
servicePeriodEndDate	servicePeriodEndDate
servicePeriodStartDate	servicePeriodStartDate
date	date
serviceFamily	serviceFamily
productOrderId	productOrderId
productOrderName	productOrderName
consumedService	consumedService
meterId	meterId
meterName	meterName
meterCategory	meterCategory
meterSubCategory	meterSubCategory
meterRegion	meterRegion
productIdIdentifier	ProductId
product	ProductName
subscriptionGuid	SubscriptionId
subscriptionName	subscriptionName
publisherType	publisherType
publisherId	publisherId

<b>Old property</b>	<b>New property</b>
publisherName	publisherName
<b>resourceGroup</b>	<b>resourceGroupName</b>
instanceName	ResourceId
<b>resourceLocationNormalized</b>	<b>location</b>
<b>resourceLocation</b>	<b>location</b>
effectivePrice	effectivePrice
quantity	quantity
unitOfMeasure	unitOfMeasure
chargeType	chargeType
<b>billingCurrencyCode</b>	<b>billingCurrency</b>
<b>pricingCurrencyCode</b>	<b>pricingCurrency</b>
costInBillingCurrency	costInBillingCurrency
costInPricingCurrency	costInPricingCurrency
costInUsd	costInUsd
paygCostInBillingCurrency	paygCostInBillingCurrency
paygCostInUSD	paygCostInUsd
exchangeRatePricingToBilling	exchangeRatePricingToBilling
exchangeRateDate	exchangeRateDate
isAzureCreditEligible	isAzureCreditEligible
serviceInfo1	serviceInfo1
serviceInfo2	serviceInfo2
additionalInfo	additionalInfo
tags	tags
partnerEarnedCreditRate	partnerEarnedCreditRate
partnerEarnedCreditApplied	partnerEarnedCreditApplied
<b>marketPrice</b>	<b>PayGPrice</b>

Old property	New property
frequency	frequency
term	term
reservationId	reservationId
reservationName	reservationName
pricingModel	pricingModel
unitPrice	unitPrice
exchangeRatePricingToBilling	exchangeRatePricingToBilling

## Next steps

- Learn more about Cost Management + Billing automation at [Cost Management automation overview](#).

# Migrate from Consumption Marketplaces API

Article • 07/17/2022

This article discusses migration away from the [Consumption Marketplaces API](#). The Consumption Marketplaces API is deprecated. The date that the API will be turned off is still being determined. We recommend that you migrate away from the API as soon as possible.

This article only applies to customers with an Enterprise Agreement or an MSDN, pay-as-you-go, or Visual Studio subscription.

## Migration destinations

We've merged Azure Marketplace and Azure usage records into a single usage details dataset. Read the [Choose a cost details solution](#) article before you choose the solution that's right for your workload. Generally, we recommend using [Exports](#) if you have ongoing data ingestion needs or a large monthly usage details dataset. For more information, see [Ingest usage details data](#).

If you have a smaller usage details dataset or a scenario that isn't met by Exports, consider using the [Cost Details](#) report instead. For more information, see [Get small cost datasets on demand](#).

### Note

The [Cost Details](#) report is only available for customers with an Enterprise Agreement or Microsoft Customer Agreement. If you have an MSDN, pay-as-you-go, or Visual Studio subscription, you can migrate to Exports or continue using the Consumption Usage Details API.

## Migration benefits

New solutions provide many benefits over the Consumption Usage Details API. Here's a summary:

- **Single dataset for all usage details** - Azure and Azure Marketplace usage details were merged into one dataset. It reduces the number of APIs that you need to call to get see all your charges.

- **Scalability** - The Marketplaces API is deprecated because it promotes a call pattern that isn't able to scale as your Azure usage increases. The usage details dataset can get exceedingly large as you deploy more resources into the cloud. The Marketplaces API is a paginated synchronous API so it isn't optimized to effectively transfer large volumes of data over a network with high efficiency and reliability. Exports and the [Cost Details](#) report are asynchronous. They provide you with a CSV file that can be directly downloaded over the network.
- **API improvements** - Exports and the Cost Details API are the solutions that Azure supports moving forward. All new features are being integrated into them.
- **Schema consistency** - The [Cost Details](#) API and [Exports](#) process provide files with matching fields so you can move from one solution to the other, based on your scenario.
- **Cost Allocation integration** - Enterprise Agreement and Microsoft Customer Agreement customers using Exports or the Cost Details API can view charges in relation to the cost allocation rules that they've configured. For more information about cost allocation, see [Allocate costs](#).

## Field differences

The following table summarizes the field mapping needed to transition from the data provided by the Marketplaces API to Exports and the Cost Details API. Both of the solutions provide a CSV file download as opposed to the paginated JSON response that's provided by the Consumption API.

Usage records can be identified as marketplace records in the combined dataset through the `PublisherType` field. Also, there are many new fields in the newer solutions that might be useful to you. For more information about available fields, see [Understand usage details fields](#).

Old Property	New Property	Notes
	PublisherType	Used to identify a marketplace usage record
accountName	AccountName	
additionalProperties	AdditionalInfo	
costCenter	CostCenter	
departmentName	BillingProfileName	
billingPeriodId		Use BillingPeriodStartDate / BillingPeriodEndDate
usageStart		Use Date

Old Property	New Property	Notes
usageEnd		Use Date
instanceName	ResourceName	
instanceId	ResourceId	
currency	BillingCurrencyCode	
consumedQuantity	Quantity	
pretaxCost	CostInBillingCurrency	
isEstimated		Not available
meterId	MeterId	
offerName	OfferId	
resourceGroup	ResourceGroup	
orderNumber		Not available
publisherName	PublisherName	
planName	PlanName	
resourceRate	EffectivePrice	
subscriptionGuid	SubscriptionId	
subscriptionName	SubscriptionName	
unitOfMeasure	UnitOfMeasure	
isRecurringCharge		Where applicable, use the Frequency and Term fields moving forward.

## Next steps

- Learn more about Cost Management automation at [Cost Management automation overview](#).

# Exports

Reference

Service: Cost Management

API Version: 2023-03-01

## Operations

<a href="#">Create Or Update</a>	The operation to create or update a export. Update operation requires latest eTag to be set in the request. You may obtain the latest eTag by performing a get o...
<a href="#">Delete</a>	The operation to delete a export.
<a href="#">Execute</a>	The operation to run an export.
<a href="#">Get</a>	The operation to get the export for the defined scope by export name.
<a href="#">Get Execution History</a>	The operation to get the run history of an export for the defined scope and export name.
<a href="#">List</a>	The operation to list all exports at the given scope.

# Generate Cost Details Report

Reference

Service: Cost Management

API Version: 2023-03-01

## Operations

<a href="#">Create Operation</a>	This API is the replacement for all previously released Usage Details APIs. Request to generate a cost details report for the provided date range, billing period...
<a href="#">Get Operation Results</a>	Get the result of the specified operation. This link is provided in the CostDetails creation request response Location header.

# Microsoft Cost Management

Article • 06/07/2022

The Cost Management APIs provide the ability to explore cost and usage data via multidimensional analysis, where creating customized filters and expressions allow you to answer consumption-related questions for your Azure resources. These APIs are currently available for Azure Enterprise customers.

## REST Operation Groups

Operation Group	Description
Dimensions	Provides operations to get supported dimensions for your usage under a variety of scopes. You can retrieve a list of dimensions that can be used as inputs for generating queries with the Query or Exports API.
Query	Provides operations to get aggregated cost and usage data based on the query you supply. Data can be filtered, sorted, and grouped by all available Dimensions (accessible through the Dimensions API).
Exports	Provides operations to schedule recurring exports of cost and usage data to blob storage. Data can be filtered, sorted, and grouped by all available Dimensions (accessible through the Dimensions API).

## See also

[Azure consumption REST API documentation](#)

# Az.CostManagement

Reference

Microsoft Azure PowerShell: Cost cmdlets

## Cost Management

<a href="#">Get-AzCostManagementExport</a>	The operation to get the export for the defined scope by export name.
<a href="#">Get-AzCostManagementExportExecutionHistory</a>	The operation to get the execution history of an export for the defined scope and export name.
<a href="#">Invoke-AzCostManagementExecuteExport</a>	The operation to execute an export.
<a href="#">Invoke-AzCostManagementQuery</a>	Query the usage data for scope defined.
<a href="#">Invoke-AzCostManagementReservationDetailReport</a>	Generates the reservations details report for provided date range asynchronously based on enrollment id. The Reservation usage details can be viewed only by certain enterprise roles. For more details on the roles see, <a href="https://learn.microsoft.com/en-us/azure/cost-management-billing/manage/understand-ea-roles#usage-and-costs-access-by-role">https://learn.microsoft.com/en-us/azure/cost-management-billing/manage/understand-ea-roles#usage-and-costs-access-by-role</a>
<a href="#">New-AzCostManagementDetailReport</a>	This API is the replacement for all previously release Usage Details APIs. Request to generate a cost details report for the provided date range, billing period (Only enterprise customers) or Invoice Id asynchronously at a certain scope. The initial call to request a report will return a 202 with a 'Location' and 'Retry-After' header. The 'Location' header will provide the endpoint to poll to get the result of the report generation. The 'Retry-After' provides the duration to wait before polling for the generated report. A call to poll the report operation will provide a 202 response with a 'Location' header if the operation

is still in progress. Once the report generation operation completes, the polling endpoint will provide a 200 response along with details on the report blob(s) that are available for download. The details on the file(s) available for download will be available in the polling response body.

<a href="#">New-AzCostManagementExport</a>	The operation to create or update a export. Update operation requires latest eTag to be set in the request. You may obtain the latest eTag by performing a get operation. Create operation does not require eTag.
<a href="#">New-AzCostManagementQueryComparisonExpressionObject</a>	Create a in-memory object for QueryComparisonExpression
<a href="#">New-AzCostManagementQueryFilterObject</a>	Create a in-memory object for QueryFilter
<a href="#">Remove-AzCostManagementExport</a>	The operation to delete a export.
<a href="#">Update-AzCostManagementExport</a>	The operation to create or update a export. Update operation requires latest eTag to be set in the request. You may obtain the latest eTag by performing a get operation. Create operation does not require eTag.

# Cost Management automation FAQ

Article • 10/12/2022

The following sections cover the most commonly asked questions and answers about Cost Management automation.

## What are Cost Details versus Usage Details?

Both are different names for the same dataset. Usage Details is the original name of the dataset back when only Azure consumption resource usage records were present. Over time, more types of cost have been added to the dataset, including marketplace usage, Azure purchases (such as Reservations), marketplace purchases and even Microsoft 365 costs. Cost Details is the name being used moving forward for the dataset.

## Why do I get Usage Details API timeouts?

Cost datasets available from the Usage Details API can often be overly large (multiple GBs or more). The larger the size of the dataset that you request, the longer the service takes to compile the data before sending it to you. Because of the delay, synchronous API solutions like the paginated [JSON Usage Details API](#) might time out before your data is provided. If you encounter timeouts or have processes that frequently need to pull a large amount of cost data, see [Retrieve large cost datasets recurrently with Exports](#).

## What is the difference between legacy and modern usage details?

A legacy versus modern usage details record is identified by the kind field in the [Usage Details API](#). The field is used to distinguish between data that's returned for different customer types. The call patterns to obtain legacy and modern usage details are essentially the same. The granularity of the data is the same. The main difference is the fields available in the usage details records themselves. If you're an EA customer, you'll always get legacy usage details records. If you're a Microsoft Customer Agreement customer, you'll always get modern usage details records.

## How do I see my recurring charges?

Recurring charges are available in the [Cost Details](#) report when viewing Actual Cost.

## Where can I see tax information in Cost Details?

Cost details data is all pre-tax. Tax related charges are only available on your invoice.

## Why is PAYGPrice zero in my cost details file?

If you're an EA customer, we don't currently support showing pay-as-you-go prices directly in the usage details data. To see the pricing, use the [Retail Prices API](#).

## Does Cost Details have Reservation charges?

Yes it does. You can see those charges according to when the actual charges occurred (Actual Cost). Or, you can see the charges spread across the resources that consumed the Reservation (Amortized Cost). For more information, see [Get Azure consumption and reservation usage data using API](#).

## Am I charged for using the Cost Details API?

No the Cost Details API is free. Make sure to abide by the rate-limiting policies, however.

## What's the difference between the Invoices API, the Transactions API, and the Cost Details API?

These APIs provide a different view of the same data:

- The [Invoices API](#) provides an aggregated view of your monthly charges.
- The [Transactions API](#) provides a view of your monthly charges aggregated at product/service family level.
- The [Cost Details](#) report provides a granular view of the usage and cost records for each day. The Cost Details API is available for Enterprise Agreement and Microsoft Customer Agreement accounts. For pay-as-you-go subscriptions, use the Exports API. If Exports don't meet your needs and you need an on-demand solution, see [Get Cost Details for a pay-as-you-go subscription](#).

## I recently migrated from an EA to an MCA agreement. How do I migrate my API workloads?

See [Migrate from EA to MCA APIs](#).

## When will the Enterprise Reporting APIs get turned off?

The Enterprise Reporting APIs are deprecated. The date that the API will be turned off is still being determined. We recommend that you migrate away from the APIs as soon as possible. For more information, see [Migrate from Azure Enterprise Reporting to Microsoft Cost Management APIs](#).

## When will the [Consumption Usage Details API](#) get turned off?

The Consumption Usage Details API is deprecated. The date that the API will be turned off is still being determined. We recommend that you migrate away from the API as soon as possible. For more information, see [Migrate from Consumption Usage Details API](#).

## When will the [Consumption Marketplaces API](#) get turned off?

The Marketplaces API is deprecated. The date that the API will be turned off is still being determined. Data from the API is available in the [Cost Details](#) report. We recommend that you migrate to it as soon as possible. For more information, see [Migrate from Consumption Marketplaces API](#).

## When will the [Consumption Forecasts API](#) get turned off?

The Forecasts API is deprecated. The date that the API will be turned off is still being determined. Data from the API is available in the [Cost Management Forecast API](#). We recommend that you migrate to it as soon as possible.

## Next steps

- Learn more about Cost Management + Billing automation at [Cost Management automation overview](#).