Azure Fundamentals Exam Prep (AZ-900)

Part 3: Describe Azure management and governance

Microsoft

Describe factors that can affect costs in Azure

Azure shifts development costs from the capital expense (CapEx) of building out and maintaining infrastructure and facilities to an operational expense (OpEx) of renting infrastructure as you need it, whether it's compute, storage, networking, and so on.

That OpEx cost can be impacted by many factors. Some of the impacting factors are:

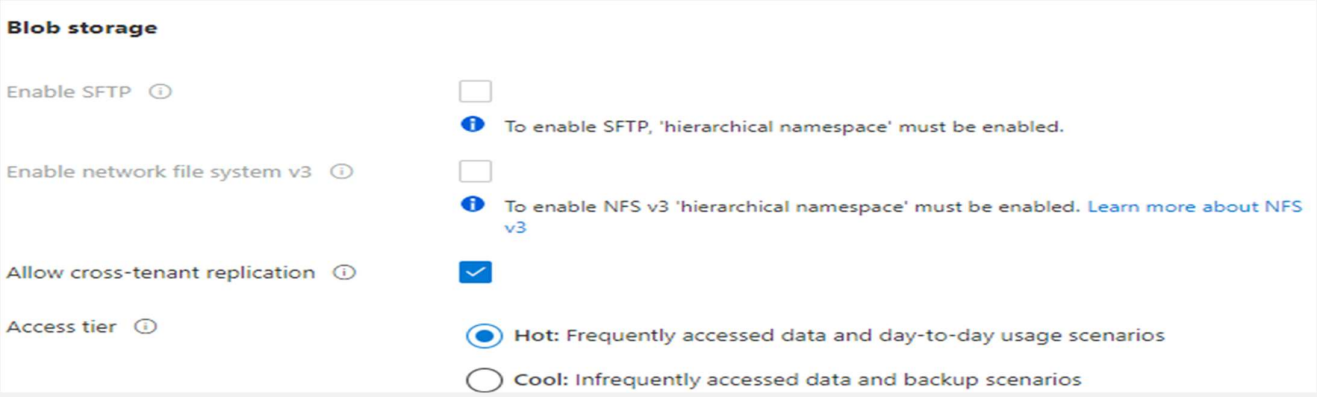
- Resource type
- Consumption
- Maintenance
- Geography
- Subscription type
- Azure Marketplace

Resource type

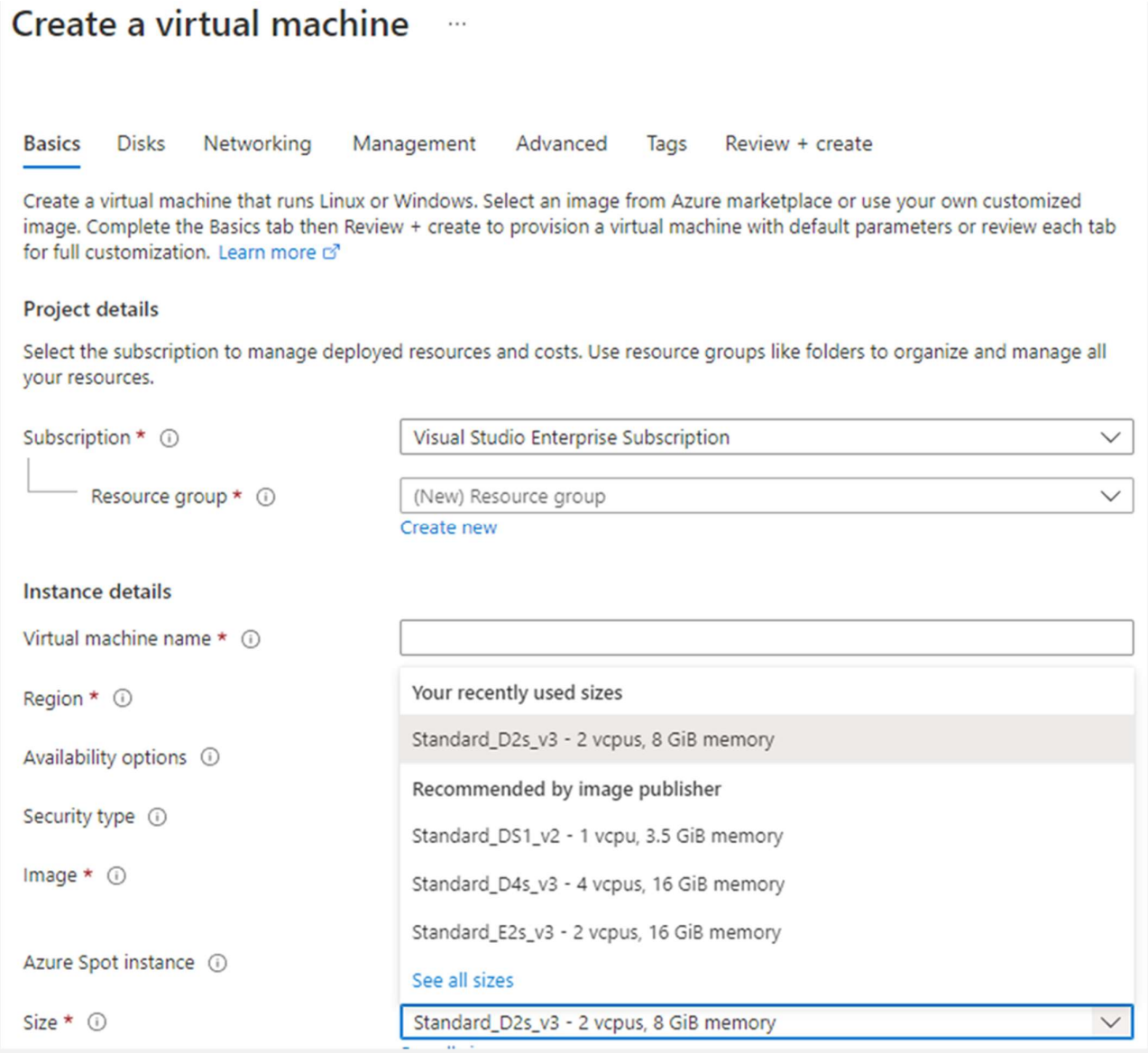
A number of factors influence the cost of Azure resources. The type of resources, the settings for the resource, and the Azure region will all have an impact on how much a resource costs. When you provision an Azure resource, Azure creates metered instances for that resource. The meters track the resources' usage and generate a usage record that is used to calculate your bill.

Examples

With a storage account, you specify a type such as blob, a performance tier, an access tier, redundancy settings, and a region. Creating the same storage account in different regions may show different costs and changing any of the settings may also impact the price.



With a virtual machine (VM), you may have to consider licensing for the operating system or other software, the processor and number of cores for the VM, the attached storage, and the network interface. Just like with storage, provisioning the same virtual machine in different regions may result in different costs.



Consumption

Pay-as-you-go has been a consistent theme throughout, and that's the cloud payment model where you pay for the resources that you use during a billing cycle. If you use more compute this cycle, you pay more. If you use less in the current cycle, you pay less. It's a straight forward pricing mechanism that allows for maximum flexibility.

However, Azure also offers the ability to commit to using a set amount of cloud resources in advance and receiving discounts on those "reserved" resources. Many services, including databases, compute, and storage all provide the option to commit to a level of use and receive a discount, in some cases up to 72 percent.

When you reserve capacity, you're committing to using and paying for a certain amount of Azure resources during a given period (typically one or three years). With the back-up of pay-as-you-go, if you see a sudden surge in demand that eclipses what you've

pre-reserved, you just pay for the additional resources in excess of your reservation. This model allows you to recognize significant savings on reliable, consistent workloads while also having the flexibility to rapidly increase your cloud footprint as the need arises.

Maintenance

The flexibility of the cloud makes it possible to rapidly adjust resources based on demand. Using resource groups can help keep all of your resources organized. In order to control costs, it's important to maintain your cloud environment. For example, every time you provision a VM, additional resources such as storage and networking are also provisioned. If you deprovision the VM, those additional resources may not deprovision at the same time, either intentionally or unintentionally. By keeping an eye on your resources and making sure you're not keeping around resources that are no longer needed, you can help control cloud costs.

Geography

When you provision most resources in Azure, you need to define a region where the resource deploys. Azure infrastructure is distributed globally, which enables you to deploy your services centrally or closest to your customers, or something in between. With this global deployment comes global pricing differences. The cost of power, labor, taxes, and fees vary depending on the location. Due to these variations, Azure resources can differ in costs to deploy depending on the region.

Network traffic is also impacted based on geography. For example, it's less expensive to move information within Europe than to move information from Europe to Asia or South America.

Network Traffic

Billing zones are a factor in determining the cost of some Azure services.

Bandwidth refers to data moving in and out of Azure datacenters. Some inbound data transfers (data going into Azure datacenters) are free. For outbound data transfers (data leaving Azure datacenters), data transfer pricing is based on zones.

A zone is a geographical grouping of Azure regions for billing purposes. The bandwidth pricing page has additional information on pricing for data ingress, egress, and transfer.

Subscription type

Some Azure subscription types also include usage allowances, which affect costs.

For example, an Azure free trial subscription provides access to a number of Azure products that are free for 12 months. It also includes credit to spend within your first 30

days of sign-up. You'll get access to more than 25 products that are always free (based on resource and region availability).

Azure Marketplace

Azure Marketplace lets you purchase Azure-based solutions and services from third-party vendors. This could be a server with software preinstalled and configured, or managed network firewall appliances, or connectors to third-party backup services. When you purchase products through Azure Marketplace, you may pay for not only the Azure services that you're using, but also the services or expertise of the third-party vendor. Billing structures are set by the vendor.

All solutions available in Azure Marketplace are certified and compliant with Azure policies and standards. The certification policies may vary based on the service or solution type and Azure service involved. Commercial marketplace certification policies has additional information on Azure Marketplace certifications.

Compare the Pricing and Total Cost of Ownership calculators

The pricing calculator and the total cost of ownership (TCO) calculator are two calculators that help you understand potential Azure expenses. Both calculators are accessible from the internet, and both calculators allow you to build out a configuration. However, the two calculators have very different purposes.

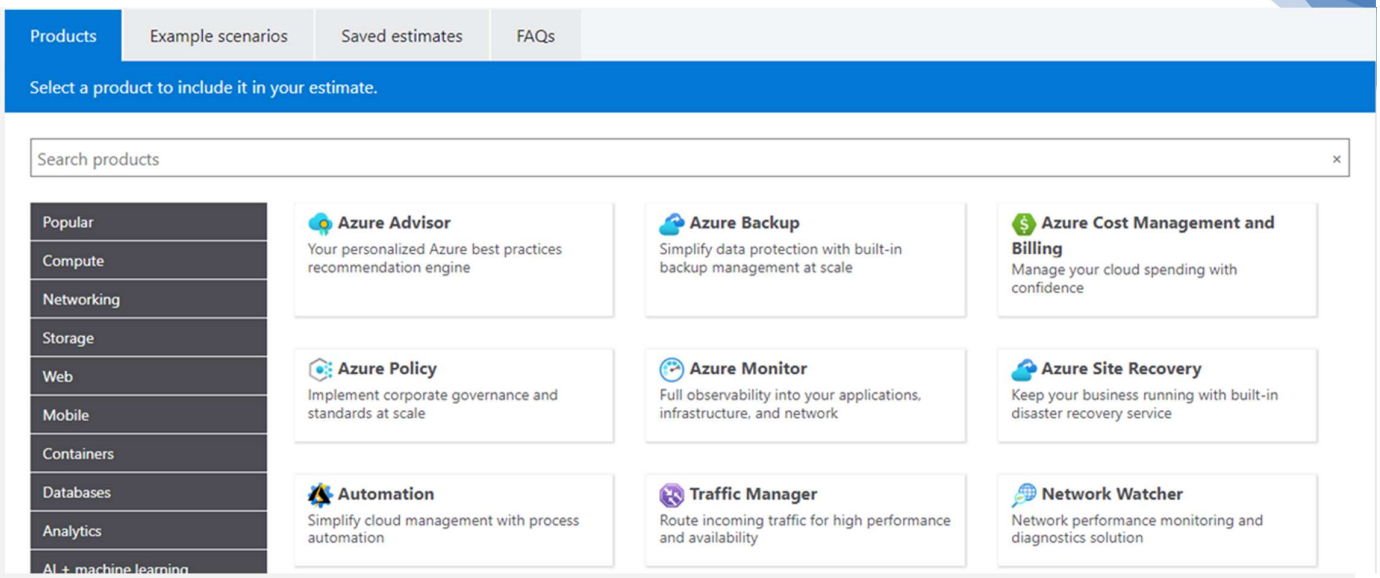
Pricing calculator

The pricing calculator is designed to give you an estimated cost for provisioning resources in Azure. You can get an estimate for individual resources, build out a solution, or use an example scenario to see an estimate of the Azure spend. The pricing calculator's focus is on the cost of provisioned resources in Azure.

Note

The Pricing calculator is for information purposes only. The prices are only an estimate. Nothing is provisioned when you add resources to the pricing calculator, and you won't be charged for any services you select.

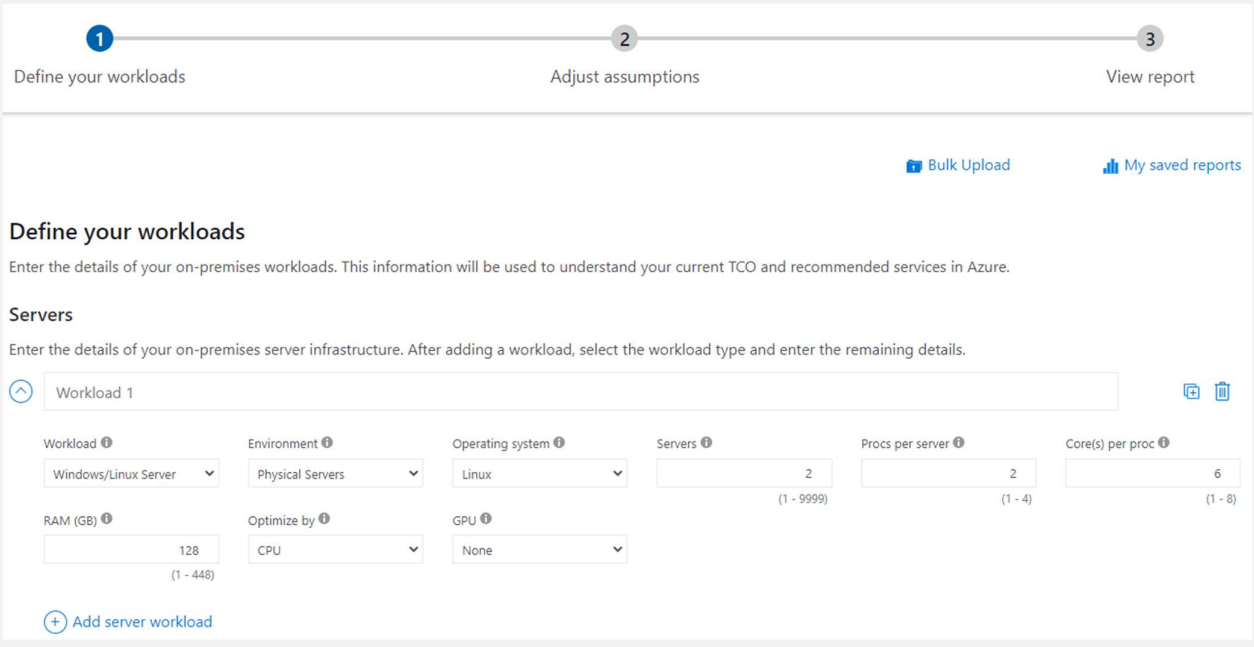
With the pricing calculator, you can estimate the cost of any provisioned resources, including compute, storage, and associated network costs. You can even account for different storage options like storage type, access tier, and redundancy.



TCO calculator

The TCO calculator is designed to help you compare the costs for running an on-premises infrastructure compared to an Azure Cloud infrastructure. With the TCO calculator, you enter your current infrastructure configuration, including servers, databases, storage, and outbound network traffic. The TCO calculator then compares the anticipated costs for your current environment with an Azure environment supporting the same infrastructure requirements.

With the TCO calculator, you enter your configuration, add in assumptions like power and IT labor costs, and are presented with an estimation of the cost difference to run the same environment in your current datacenter or in Azure.



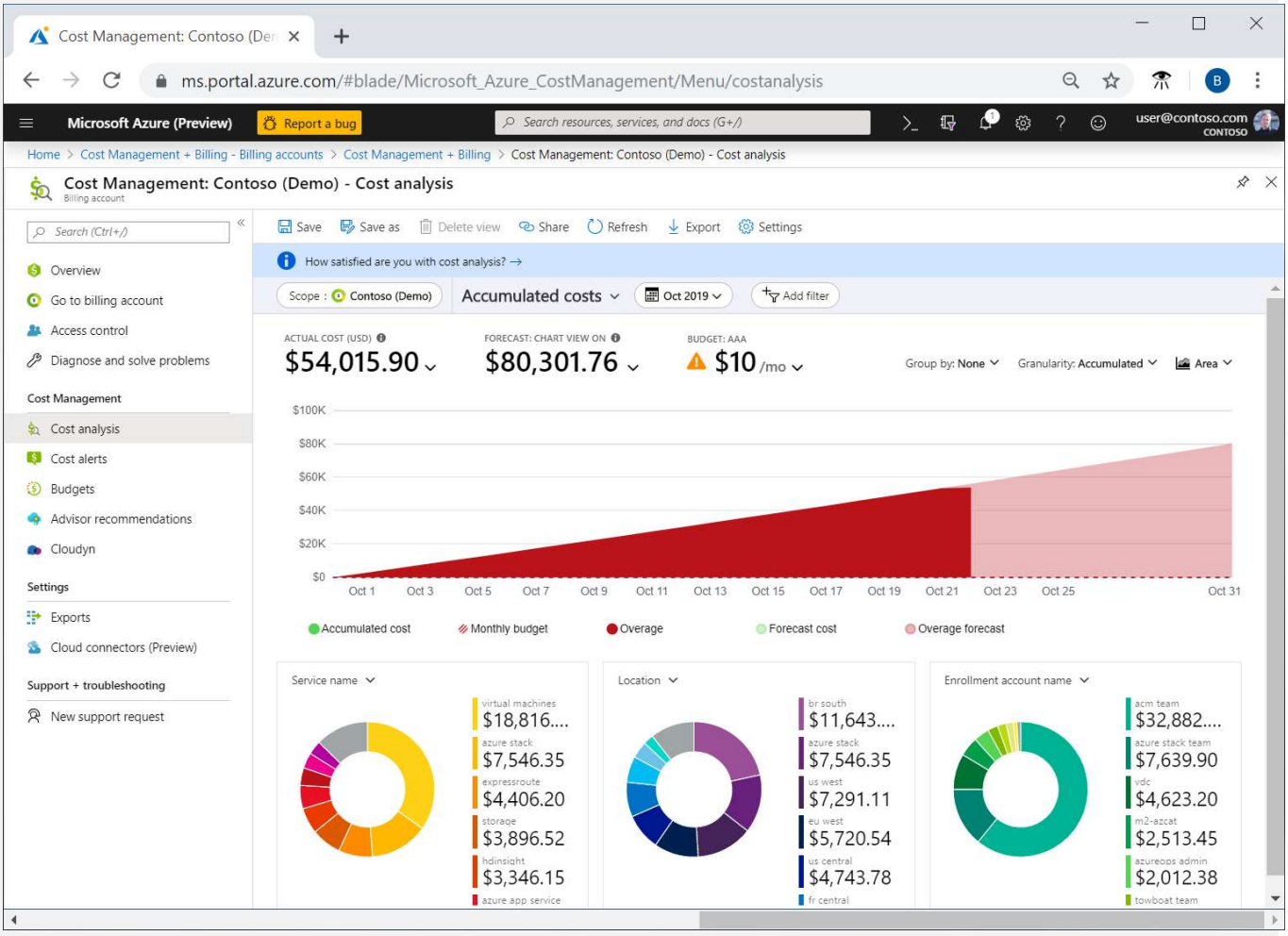
Describe the Azure Cost Management tool

Microsoft Azure is a global cloud provider, meaning you can provision resources anywhere in the world. You can provision resources rapidly to meet a sudden demand, or to test out a new feature, or on accident. If you accidentally provision new resources, you may not be aware of them until it's time for your invoice. Cost Management is an Azure service that helps avoid those situations.

What is Cost Management?

Cost Management provides the ability to quickly check Azure resource costs, create alerts based on resource spend, and create budgets that can be used to automate management of resources.

Cost analysis is a subset of Cost Management that provides a quick visual for your Azure costs. Using cost analysis, you can quickly view the total cost in a variety of different ways, including by billing cycle, region, resource, and so on.



You use cost analysis to explore and analyze your organizational costs. You can view aggregated costs by organization to understand where costs are accrued and to identify

spending trends. And you can see accumulated costs over time to estimate monthly, quarterly, or even yearly cost trends against a budget.

Cost alerts

Cost alerts provide a single location to quickly check on all of the different alert types that may show up in the Cost Management service. The three types of alerts that may show up are:

- Budget alerts
- Credit alerts
- Department spending quota alerts.

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. Budgets are defined by cost or by consumption usage when using the Azure Consumption API. 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 appears in cost alerts. An alert email is also sent to the people in the alert recipients list of the budget.

Credit alerts

Credit alerts notify you when your Azure credit monetary commitments are consumed. Monetary commitments are for organizations with Enterprise Agreements (EAs). Credit alerts are generated automatically at 90% and at 100% of your Azure 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 appears in cost alerts. For example, 50 percent or 75 percent of the quota.

Budgets

A budget is where you set a spending limit for Azure. You can set budgets based on a subscription, resource group, service type, or other criteria. When you set a budget, you will also set a budget alert. When the budget hits the budget alert level, it will trigger a

budget alert that shows up in the cost alerts area. If configured, budget alerts will also send an email notification that a budget alert threshold has been triggered.

A more advanced use of budgets enables budget conditions to trigger automation that suspends or otherwise modifies resources once the trigger condition has occurred.

Describe the purpose of tags

As your cloud usage grows, it's increasingly important to stay organized. A good organization strategy helps you understand your cloud usage and can help you manage costs.

One way to organize related resources is to place them in their own subscriptions. You can also use resource groups to manage related resources. Resource tags are another way to organize resources. Tags provide extra information, or metadata, about your resources. This metadata is useful for:

- **Resource management** Tags enable you to locate and act on resources that are associated with specific workloads, environments, business units, and owners.
- **Cost management and optimization** Tags enable you to group resources so that you can report on costs, allocate internal cost centers, track budgets, and forecast estimated cost.
- **Operations management** Tags enable you to group resources according to how critical their availability is to your business. This grouping helps you formulate service-level agreements (SLAs). An SLA is an uptime or performance guarantee between you and your users.
- **Security** Tags enable you to classify data by its security level, such as public or confidential.
- **Governance and regulatory compliance** Tags enable you to identify resources that align with governance or regulatory compliance requirements, such as ISO 27001. Tags can also be part of your standards enforcement efforts. For example, you might require that all resources be tagged with an owner or department name.
- **Workload optimization and automation** Tags can help you visualize all of the resources that participate in complex deployments. For example, you might tag a resource with its associated workload or application name and use software such as Azure DevOps to perform automated tasks on those resources.

How do I manage resource tags?

You can add, modify, or delete resource tags through Windows PowerShell, the Azure CLI, Azure Resource Manager templates, the REST API, or the Azure portal.

You can use Azure Policy to enforce tagging rules and conventions. For example, you can require that certain tags be added to new resources as they're provisioned. You can also define rules that reapply tags that have been removed. Resources don't inherit tags from subscriptions and resource groups, meaning that you can apply tags at one level and not have those tags automatically show up at a different level, allowing you to

create custom tagging schemas that change depending on the level (resource, resource group, subscription, and so on).

Describe the purpose of Azure Blueprints

What happens when your cloud starts to grow beyond just one subscription or environment? How can you scale the configuration of features? How can you enforce settings and policies in new subscriptions?

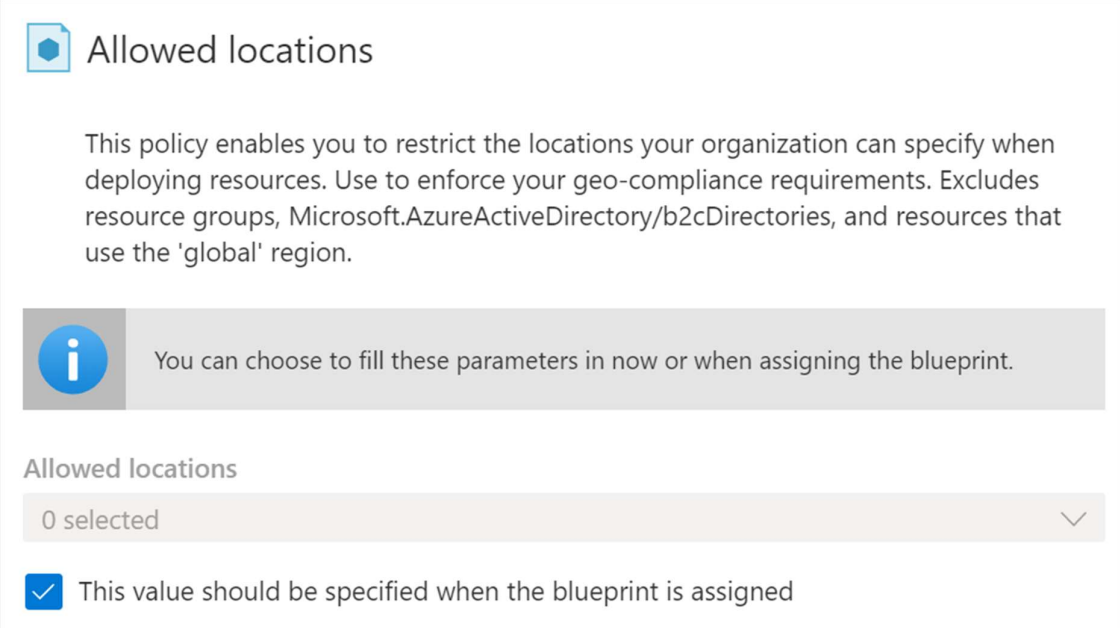
Azure Blueprints lets you standardize cloud subscription or environment deployments. Instead of having to configure features like Azure Policy for each new subscription, with Azure Blueprints you can define repeatable settings and policies that are applied as new subscriptions are created. Need a new test/dev environment? Azure Blueprints lets you deploy a new Test/Dev environment with security and compliance settings already configured. In this way, development teams can rapidly build and deploy new environments with the knowledge that they're building within organizational requirements.

What are artifacts?

Each component in the blueprint definition is known as an artifact.

It is possible for artifacts to have no additional parameters (configurations). An example is the Deploy threat detection on SQL servers policy, which requires no additional configuration.

Artifacts can also contain one or more parameters that you can configure. The following screenshot shows the Allowed locations policy. This policy includes a parameter that specifies the allowed locations.



You can specify a parameter's value when you create the blueprint definition or when you assign the blueprint definition to a scope. In this way, you can maintain one standard blueprint but have the flexibility to specify the relevant configuration parameters at each scope where the definition is assigned.

Azure Blueprints deploy a new environment based on all of the requirements, settings, and configurations of the associated artifacts. Artifacts can include things such as:

- Role assignments
- Policy assignments
- Azure Resource Manager templates
- Resource groups

How do Azure Blueprints help monitor deployments?

Azure Blueprints are version-able, allowing you to create an initial configuration and then make updates later on and assign a new version to the update. With versioning, you can make small updates and keep track of which deployments used which configuration set.

With Azure Blueprints, the relationship between the blueprint definition (what should be deployed) and the blueprint assignment (what was deployed) is preserved. In other words, Azure creates a record that associates a resource with the blueprint that defines it. This connection helps you track and audit your deployments.

Describe the purpose of Azure Policy

How do you ensure that your resources stay compliant? Can you be alerted if a resource's configuration has changed?

Azure Policy is a service in Azure that enables you to create, assign, and manage policies that control or audit your resources. These policies enforce different rules across your resource configurations so that those configurations stay compliant with corporate standards.

How does Azure Policy define policies?

Azure Policy enables you to define both individual policies and groups of related policies, known as initiatives. Azure Policy evaluates your resources and highlights resources that aren't compliant with the policies you've created. Azure Policy can also prevent noncompliant resources from being created.

Azure Policies can be set at each level, enabling you to set policies on a specific resource, resource group, subscription, and so on. Additionally, Azure Policies are inherited, so if you set a policy at a high level, it will automatically be applied to all of the groupings that fall within the parent. For example, if you set an Azure Policy on a

resource group, all resources created within that resource group will automatically receive the same policy.

Azure Policy comes with built-in policy and initiative definitions for Storage, Networking, Compute, Security Center, and Monitoring. For example, if you define a policy that allows only a certain size for the virtual machines (VMs) to be used in your environment, that policy is invoked when you create a new VM and whenever you resize existing VMs. Azure Policy also evaluates and monitors all current VMs in your environment, including VMs that were created before the policy was created.

In some cases, Azure Policy can automatically remediate noncompliant resources and configurations to ensure the integrity of the state of the resources. For example, if all resources in a certain resource group should be tagged with AppName tag and a value of "SpecialOrders," Azure Policy will automatically apply that tag if it is missing. However, you still retain full control of your environment. If you have a specific resource that you don't want Azure Policy to automatically fix, you can flag that resource as an exception – and the policy won't automatically fix that resource.

Azure Policy also integrates with Azure DevOps by applying any continuous integration and delivery pipeline policies that pertain to the pre-deployment and post-deployment phases of your applications.

What are Azure Policy initiatives?

An Azure Policy initiative is a way of grouping related policies together. The initiative definition contains all of the policy definitions to help track your compliance state for a larger goal.

For example, Azure Policy includes an initiative named Enable Monitoring in Azure Security Center. Its goal is to monitor all available security recommendations for all Azure resource types in Azure Security Center.

Under this initiative, the following policy definitions are included:

- **Monitor unencrypted SQL Database in Security Center** This policy monitors for unencrypted SQL databases and servers.
- **Monitor OS vulnerabilities in Security Center** This policy monitors servers that don't satisfy the configured OS vulnerability baseline.
- **Monitor missing Endpoint Protection in Security Center** This policy monitors for servers that don't have an installed endpoint protection agent.

In fact, the Enable Monitoring in Azure Security Center initiative contains over 100 separate policy definitions.

Describe the purpose of resource locks

A resource lock prevents resources from being accidentally deleted or changed.

Even with Azure role-based access control (Azure RBAC) policies in place, there's still a risk that people with the right level of access could delete critical cloud resources. Resource locks prevent resources from being deleted or updated, depending on the type of lock. Resource locks can be applied to individual resources, resource groups, or even an entire subscription. Resource locks are inherited, meaning that if you place a resource lock on a resource group, all of the resources within the resource group will also have the resource lock applied.

Types of Resource Locks

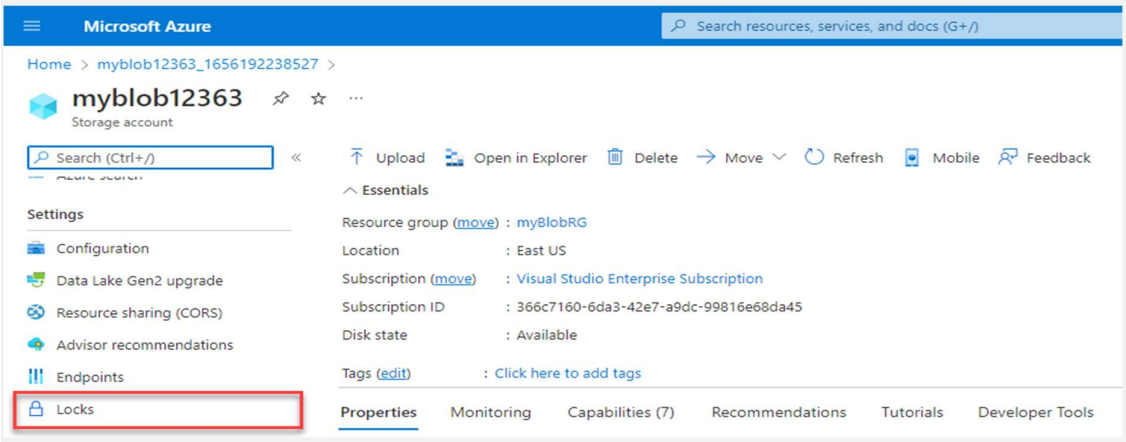
There are two types of resource locks, one that prevents users from deleting and one that prevents users from changing or deleting a resource.

- Delete means authorized users can still read and modify a resource, but they can't delete the resource.
- ReadOnly means authorized users can read a resource, but they can't delete or update the resource. Applying this lock is similar to restricting all authorized users to the permissions granted by the Reader role.

How do I manage resource locks?

You can manage resource locks from the Azure portal, PowerShell, the Azure CLI, or from an Azure Resource Manager template.

To view, add, or delete locks in the Azure portal, go to the Settings section of any resource's Settings pane in the Azure portal.



How do I delete or change a locked resource?

Although locking helps prevent accidental changes, you can still make changes by following a two-step process.

To modify a locked resource, you must first remove the lock. After you remove the lock, you can apply any action you have permissions to perform. Resource locks apply regardless of RBAC permissions. Even if you're an owner of the resource, you must still remove the lock before you can perform the blocked activity.

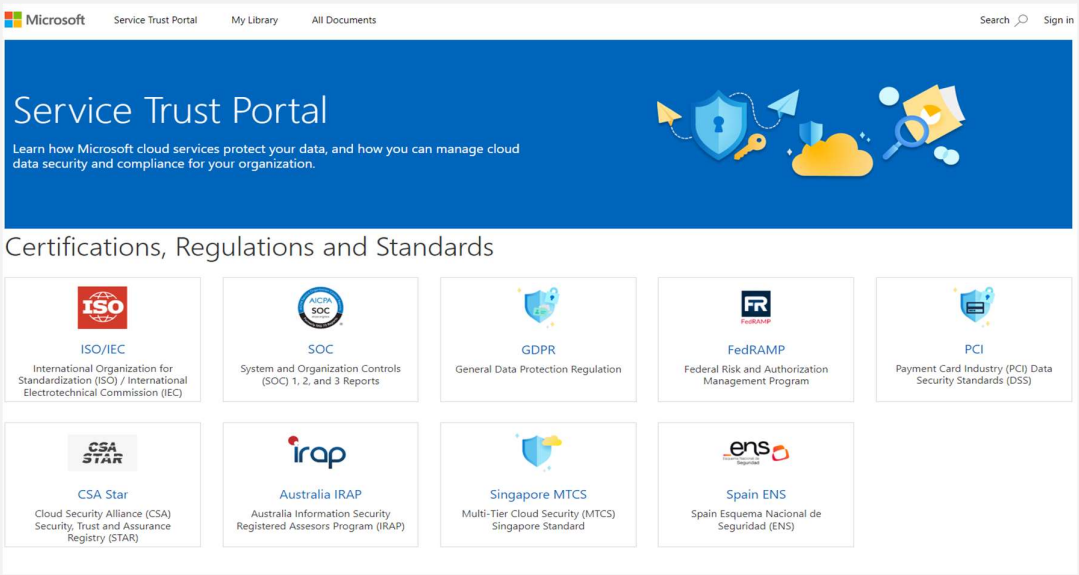
Describe the purpose of the Service Trust portal

The Microsoft Service Trust Portal is a portal that provides access to various content, tools, and other resources about Microsoft security, privacy, and compliance practices.

The Service Trust Portal contains details about Microsoft's implementation of controls and processes that protect our cloud services and the customer data therein. To access some of the resources on the Service Trust Portal, you must sign in as an authenticated user with your Microsoft cloud services account (Azure Active Directory organization account). You'll need to review and accept the Microsoft non-disclosure agreement for compliance materials.

Accessing the Service Trust Portal

You can access the Service Trust Portal at https://servicetrust.microsoft.com/.



The Service Trust Portal features and content are accessible from the main menu. The categories on the main menu are:

- **Service Trust Portal** provides a quick access hyperlink to return to the Service Trust Portal home page.

- **My Library** lets you save (or pin) documents to quickly access them on your My Library page. You can also set up to receive notifications when documents in your My Library are updated.
- **All Documents** is a single landing place for documents on the service trust portal. From **All Documents**, you can pin documents to have them show up in your **My Library**.

Note

Service Trust Portal reports and documents are available to download for at least 12 months after publishing or until a new version of document becomes available.

Describe tools for interacting with Azure

To get the most out of Azure, you need a way to interact with the Azure environment, the management groups, subscriptions, resource groups, resources, and so on. Azure provides multiple tools for managing your environment, including the:

- Azure portal
- Azure PowerShell
- Azure Command Line Interface (CLI)

What is the Azure portal?

The Azure portal is a web-based, unified console that provides an alternative to command-line tools. With the Azure portal, you can manage your Azure subscription by using a graphical user interface. You can:

- Build, manage, and monitor everything from simple web apps to complex cloud deployments
- Create custom dashboards for an organized view of resources
- Configure accessibility options for an optimal experience

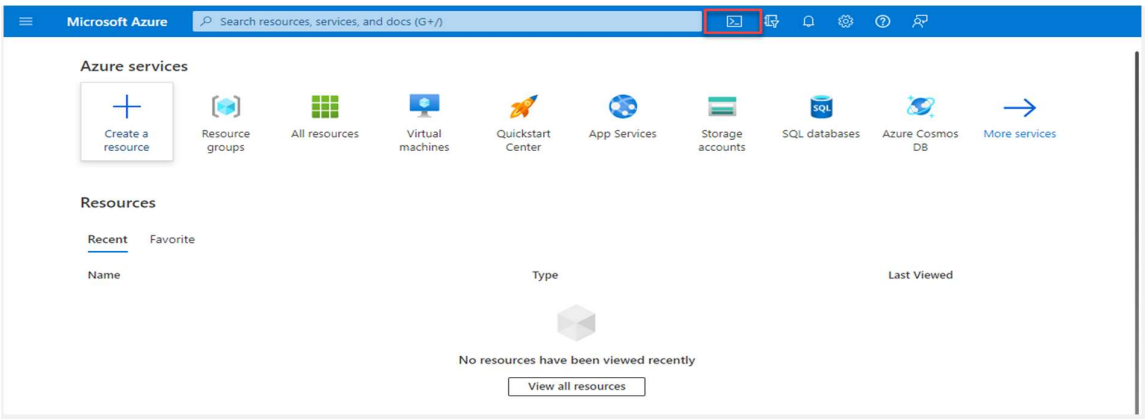
The following video introduces you to the Azure portal:

The Azure portal is designed for resiliency and continuous availability. It maintains a presence in every Azure datacenter. This configuration makes the Azure portal resilient to individual datacenter failures and avoids network slowdowns by being close to users. The Azure portal updates continuously and requires no downtime for maintenance activities.

Azure Cloud Shell

Azure Cloud Shell is a browser-based shell tool that allows you to create, configure, and manage Azure resources using a shell. Azure Cloud Shell support both Azure PowerShell and the Azure Command Line Interface (CLI), which is a Bash shell.

You can access Azure Cloud Shell via the Azure portal by selecting the Cloud Shell icon:



Azure Cloud Shell has several features that make it a unique offering to support you in managing Azure. Some of those features are:

- It is a browser-based shell experience, with no local installation or configuration required.
- It is authenticated to your Azure credentials, so when you log in it inherently knows who you are and what permissions you have.
- You choose the shell you're most familiar with; Azure Cloud Shell supports both Azure PowerShell and the Azure CLI (which uses Bash).

What is Azure PowerShell?

Azure PowerShell is a shell with which developers, DevOps, and IT professionals can run commands called command-lets (cmdlets). These commands call the Azure REST API to perform management tasks in Azure. Cmdlets can be run independently to handle one-off changes, or they may be combined to help orchestrate complex actions such as:

- The routine setup, teardown, and maintenance of a single resource or multiple connected resources.
- The deployment of an entire infrastructure, which might contain dozens or hundreds of resources, from imperative code.

Capturing the commands in a script makes the process repeatable and automatable.

In addition to be available via Azure Cloud Shell, you can install and configure Azure PowerShell on Windows, Linux, and Mac platforms.

What is the Azure CLI?

The Azure CLI is functionally equivalent to Azure PowerShell, with the primary difference being the syntax of commands. While Azure PowerShell uses PowerShell commands, the Azure CLI uses Bash commands.

The Azure CLI provides the same benefits of handling discrete tasks or orchestrating complex operations through code. It's also installable on Windows, Linux, and Mac platforms, as well as through Azure Cloud Shell.

Due to the similarities in capabilities and access between Azure PowerShell and the Bash based Azure CLI, it mainly comes down to which language you're most familiar with.

Describe the purpose of Azure Arc

Managing hybrid and multi-cloud environments can rapidly get complicated. Azure provides a host of tools to provision, configure, and monitor Azure resources. What about the on-premises resources in a hybrid configuration or the cloud resources in a multi-cloud configuration?

In utilizing Azure Resource Manager (ARM), Arc lets you extend your Azure compliance and monitoring to your hybrid and multi-cloud configurations. Azure Arc simplifies governance and management by delivering a consistent multi-cloud and on-premises management platform.

Azure Arc provides a centralized, unified way to:

- Manage your entire environment together by projecting your existing non-Azure resources into ARM.
- Manage multi-cloud and hybrid virtual machines, Kubernetes clusters, and databases as if they are running in Azure.
- Use familiar Azure services and management capabilities, regardless of where they live.
- Continue using traditional ITOps while introducing DevOps practices to support new cloud and native patterns in your environment.
- Configure custom locations as an abstraction layer on top of Azure Arc-enabled Kubernetes clusters and cluster extensions.

What can Azure Arc do outside of Azure?

Currently, Azure Arc allows you to manage the following resource types hosted outside of Azure:

- Servers
- Kubernetes clusters
- Azure data services
- SQL Server
- Virtual machines (preview)

Describe Azure Resource Manager and Azure ARM templates

Azure Resource Manager (ARM) is the deployment and management service for Azure. It provides a management layer that enables you to create, update, and delete resources in your Azure account. Anytime you do anything with your Azure resources, ARM is involved.

When a user sends a request from any of the Azure tools, APIs, or SDKs, ARM receives the request. ARM authenticates and authorizes the request. Then, ARM sends the request to the Azure service, which takes the requested action. You see consistent results and capabilities in all the different tools because all requests are handled through the same API.

Azure Resource Manager benefits

With Azure Resource Manager, you can:

- Manage your infrastructure through declarative templates rather than scripts. A Resource Manager template is a JSON file that defines what you want to deploy to Azure.
- Deploy, manage, and monitor all the resources for your solution as a group, rather than handling these resources individually.
- Re-deploy your solution throughout the development life-cycle and have confidence your resources are deployed in a consistent state.
- Define the dependencies between resources, so they're deployed in the correct order.
- Apply access control to all services because RBAC is natively integrated into the management platform.
- Apply tags to resources to logically organize all the resources in your subscription.
- Clarify your organization's billing by viewing costs for a group of resources that share the same tag.

The following video provides an overview of how you can use different Azure tools with ARM to manage your environment:

ARM templates

Infrastructure as code is a concept where you manage your infrastructure as lines of code. Leveraging Azure Cloud Shell, Azure PowerShell, or the Azure CLI are some examples of using code to deploy cloud infrastructure. ARM templates are another example of infrastructure as code at work.

By using ARM templates, you can describe the resources you want to use in a declarative JSON format. With an ARM template, the deployment code is verified before any code is run. This ensures that the resources will be created and connected correctly. The template then orchestrates the creation of those resources in parallel. That is, if you need 50 instances of the same resource, all 50 instances are created at the same time.

Ultimately, the developer, DevOps professional, or IT professional needs only to define the desired state and configuration of each resource in the ARM template, and the template does the rest. Templates can even execute PowerShell and Bash scripts before or after the resource has been set up.

Benefits of using ARM templates

ARM templates provide many benefits when planning for deploying Azure resources. Some of those benefits include:

- **Declarative syntax**: ARM templates allow you to create and deploy an entire Azure infrastructure declaratively. Declarative syntax means you declare what you want to deploy but don't need to write the actual programming commands and sequence to deploy the resources.
- **Repeatable results**: Repeatedly deploy your infrastructure throughout the development lifecycle and have confidence your resources are deployed in a consistent manner. You can use the same ARM template to deploy multiple dev/test environments, knowing that all the environments are the same.
- **Orchestration**: You don't have to worry about the complexities of ordering operations. Azure Resource Manager orchestrates the deployment of interdependent resources, so they're created in the correct order. When possible, Azure Resource Manager deploys resources in parallel, so your deployments finish faster than serial deployments. You deploy the template through one command, rather than through multiple imperative commands.
- **Modular files**: You can break your templates into smaller, reusable components and link them together at deployment time. You can also nest one template inside another template. For example, you could create a template for a VM stack, and then nest that template inside of templates that deploy entire environments, and that VM stack will consistently be deployed in each of the environment templates.
- **Extensibility**: With deployment scripts, you can add PowerShell or Bash scripts to your templates. The deployment scripts extend your ability to set up resources during deployment. A script can be included in the template or stored in an external source and referenced in the template. Deployment scripts give you the ability to complete your end-to-end environment setup in a single ARM template.

Describe the purpose of Azure Advisor

Azure Advisor evaluates your Azure resources and makes recommendations to help improve reliability, security, and performance, achieve operational excellence, and reduce costs. Azure Advisor is designed to help you save time on cloud optimization. The recommendation service includes suggested actions you can take right away, postpone, or dismiss.

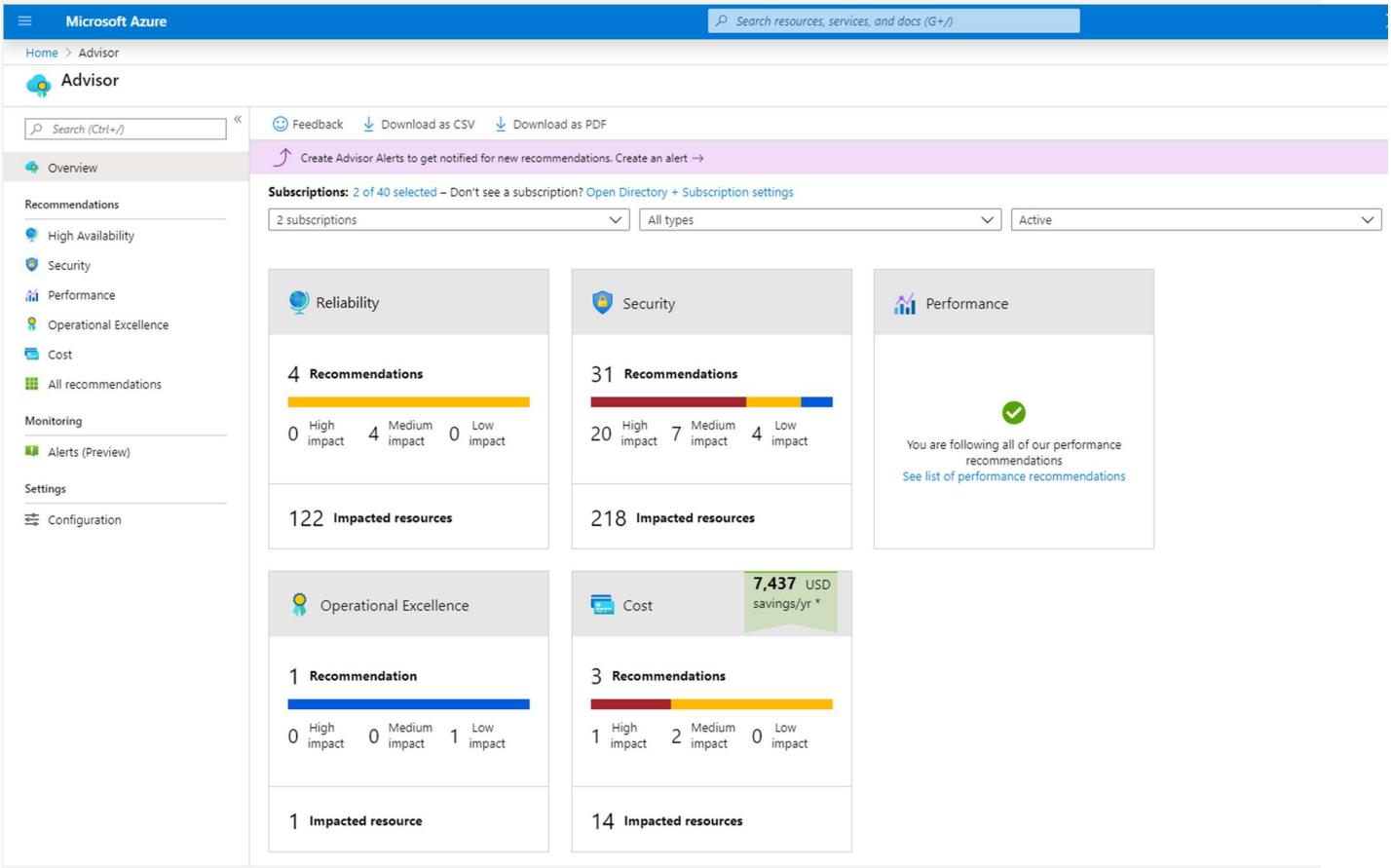
The recommendations are available via the Azure portal and the API, and you can set up notifications to alert you to new recommendations.

When you're in the Azure portal, the Advisor dashboard displays personalized recommendations for all your subscriptions. You can use filters to select

recommendations for specific subscriptions, resource groups, or services. The recommendations are divided into five categories:

- **Reliability** is used to ensure and improve the continuity of your business-critical applications.
- **Security** is used to detect threats and vulnerabilities that might lead to security breaches.
- **Performance** is used to improve the speed of your applications.
- **Operational Excellence** is used to help you achieve process and workflow efficiency, resource manageability, and deployment best practices.
- **Cost** is used to optimize and reduce your overall Azure spending.

The following image shows the Azure Advisor dashboard.



Describe Azure Service Health

Microsoft Azure provides a global cloud solution to help you manage your infrastructure needs, reach your customers, innovate, and adapt rapidly. Knowing the status of the global Azure infrastructure and your individual resources could seem like a daunting task. Azure Service Health helps you keep track of Azure resource, both your specifically deployed resources and the overall status of Azure. Azure service health does this by combining three different Azure services:

- **Azure Status** is a broad picture of the status of Azure globally. Azure status informs you of service outages in Azure on the Azure Status page. The page is a global view of the health of all Azure services across all Azure regions. It's a good reference for incidents with widespread impact.
- **Service Health** provides a narrower view of Azure services and regions. It focuses on the Azure services and regions you're using. This is the best place to look for service impacting communications about outages, planned maintenance activities, and other health advisories because the authenticated Service Health experience knows which services and resources you currently use. You can even set up Service Health alerts to notify you when service issues, planned maintenance, or other changes may affect the Azure services and regions you use.
- **Resource Health** is a tailored view of your actual Azure resources. It provides information about the health of your individual cloud resources, such as a specific virtual machine instance. Using Azure Monitor, you can also configure alerts to notify you of availability changes to your cloud resources.

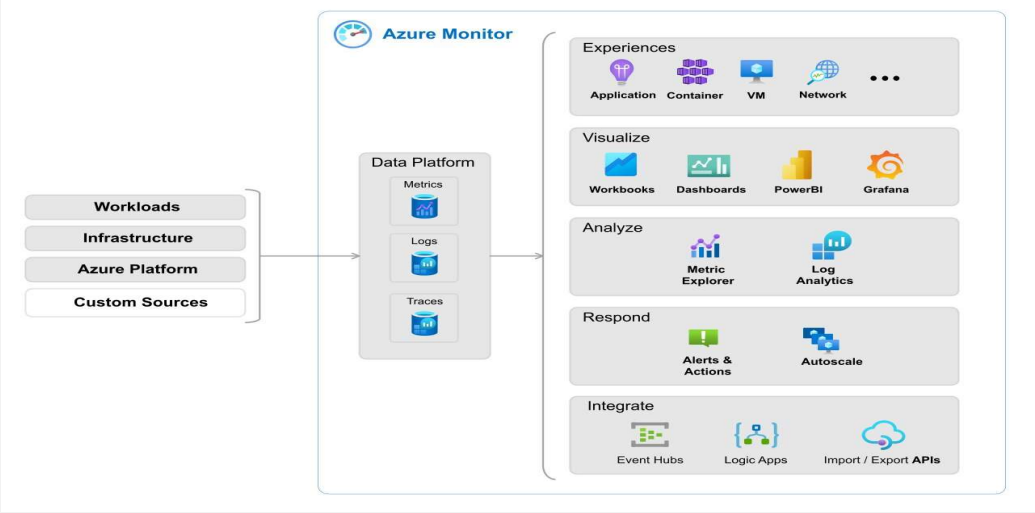
By using Azure status, Service health, and Resource health, Azure Service Health gives you a complete view of your Azure environment-all the way from the global status of Azure services and regions down to specific resources. Additionally, historical alerts are stored and accessible for later review. Something you initially thought was a simple anomaly that turned into a trend, can readily be reviewed and investigated thanks to the historical alerts.

Finally, in the event that a workload you're running is impacted by an event, Azure Service Health provides links to support.

Describe Azure Monitor

Azure Monitor is a platform for collecting data on your resources, analyzing that data, visualizing the information, and even acting on the results. Azure Monitor can monitor Azure resources, your on-premises resources, and even multi-cloud resources like virtual machines hosted with a different cloud provider.

The following diagram illustrates just how comprehensive Azure Monitor is:



On the left is a list of the sources of logging and metric data that can be collected at every layer in your application architecture, from application to operating system and network.

In the center, the logging and metric data are stored in central repositories.

On the right, the data is used in several ways. You can view real-time and historical performance across each layer of your architecture or aggregated and detailed information. The data is displayed at different levels for different audiences. You can view high-level reports on the Azure Monitor Dashboard or create custom views by using Power BI and Kusto queries.

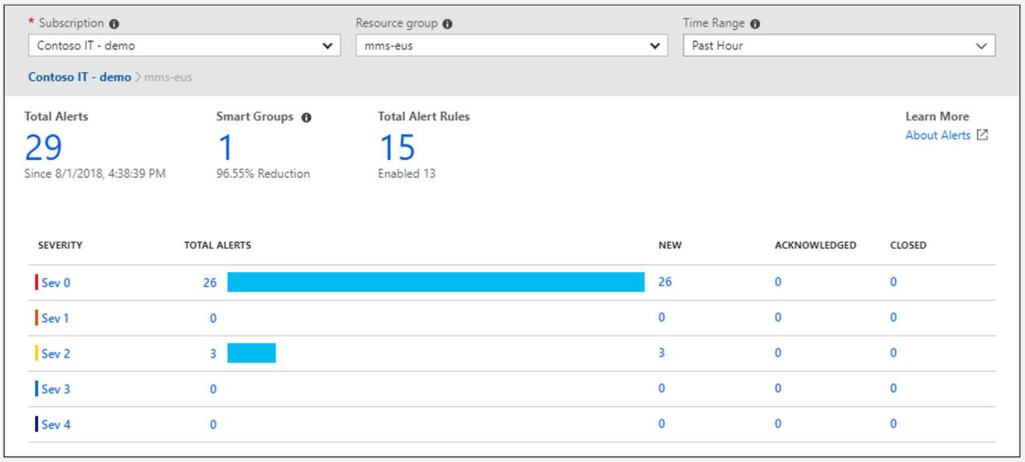
Additionally, you can use the data to help you react to critical events in real time, through alerts delivered to teams via SMS, email, and so on. Or you can use thresholds to trigger autoscaling functionality to scale to meet the demand.

Azure Log Analytics

Azure Log Analytics is the tool in the Azure portal where you'll write and run log queries on the data gathered by Azure Monitor. Log Analytics is a robust tool that supports both simple, complex queries, and data analysis. You can write a simple query that returns a set of records and then use features of Log Analytics to sort, filter, and analyze the records. You can write an advanced query to perform statistical analysis and visualize the results in a chart to identify a particular trend. Whether you work with the results of your queries interactively or use them with other Azure Monitor features such as log query alerts or workbooks, Log Analytics is the tool that you're going to use to write and test those queries.

Azure Monitor Alerts

Azure Monitor Alerts are an automated way to stay informed when Azure Monitor detects a threshold being crossed. You set the alert conditions, the notification actions, and then Azure Monitor Alerts notifies when an alert is triggered. Depending on your configuration, Azure Monitor Alerts can also attempt corrective action.



Alerts can be set up to monitor the logs and trigger on certain log events, or they can be set to monitor metrics and trigger when certain metrics are crossed. For example, you could set a metric-based alert up to notify you when the CPU usage on a virtual machine exceeded 80%. Alert rules based on metrics provide near real time alerts based on numeric values. Rules based on logs allow for complex logic across data from multiple sources.

Azure Monitor Alerts use action groups to configure who to notify and what action to take. An action group is simply a collection of notification and action preferences that you associate with one or multiple alerts. Azure Monitor, Service Health, and Azure Advisor all use actions groups to notify you when an alert has been triggered.

Application Insights

Application Insights, an Azure Monitor feature, monitors your web applications. Application Insights is capable of monitoring applications that are running in Azure, on-premises, or in a different cloud environment.

There are two ways to configure Application Insights to help monitor your application. You can either install an SDK in your application, or you can use the Application Insights agent. The Application Insights agent is supported in C#.NET, VB.NET, Java, JavaScript, Node.js, and Python.

Once Application Insights is up and running, you can use it to monitor a broad array of information, such as:

- Request rates, response times, and failure rates
- Dependency rates, response times, and failure rates, to show whether external services are slowing down performance
- Page views and load performance reported by users' browsers
- AJAX calls from web pages, including rates, response times, and failure rates
- User and session counts
- Performance counters from Windows or Linux server machines, such as CPU, memory, and network usage

Not only does Application Insights help you monitor the performance of your application, but you can also configure it to periodically send synthetic requests to your application, allowing you to check the status and monitor your application even during periods of low activity.