**PART 3: Describe Azure management and governance (30–35%)**

# 3.1: Describe cost management in Azure

## Learning objectives

1. Describe factors that can affect costs in Azure
2. Compare the Pricing calculator and Total Cost of Ownership (TCO) calculator
3. Describe the Microsoft Cost Management tool
4. Describe the purpose of tags

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** : 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.
* **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. Azure also offers the ability to commit to using a set amount of cloud resources in advance and receiving discounts on those **“reserved” resources**. (typically one or three years)
* **Maintenance:**
* **Geography:** The cost of power, labor, taxes, and fees vary depending on the location. 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.

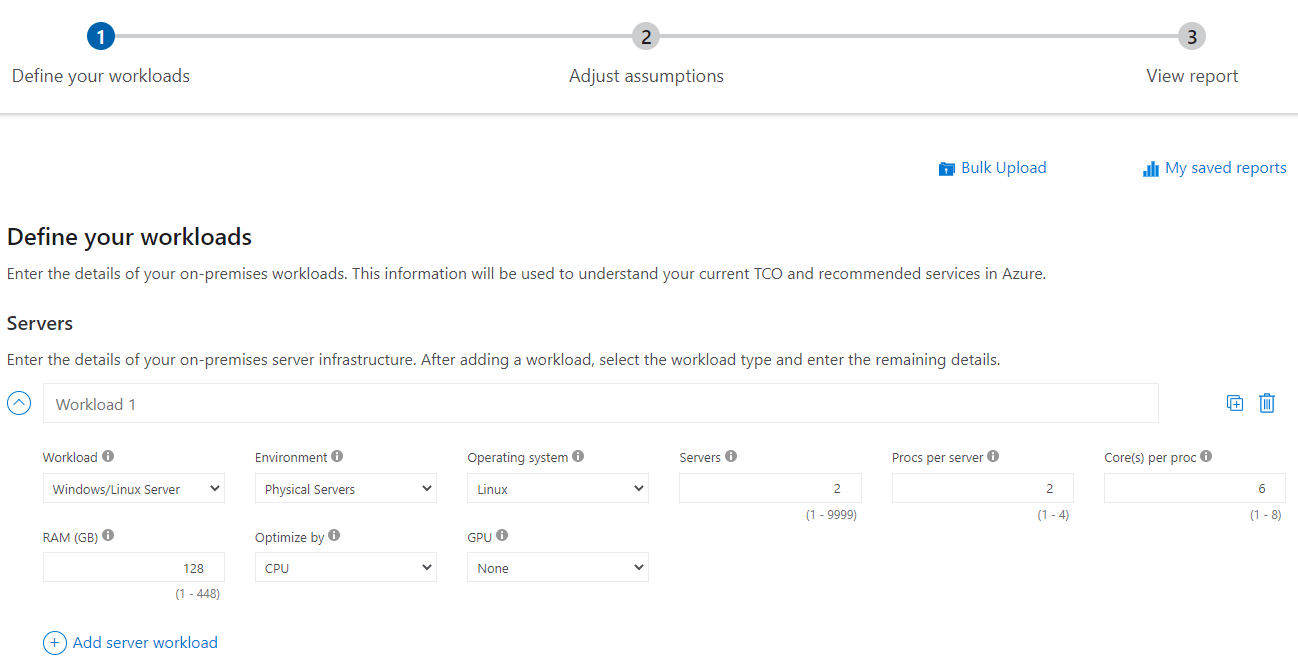
A **zone** is a geographical grouping of Azure regions for billing purposes.

* **Subscription type:** 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. All solutions available in Azure Marketplace are certified and compliant with Azure policies and standards

**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.

The pricing calculator is designed to give you an estimated cost for provisioning resources in Azure. 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.

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.



If you accidentally provision new resources, you may not be aware of them until it’s time for your invoice. **Cost Management** is a service that helps avoid those situations.

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. 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** 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:

1. **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

1. **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.

1. **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.

**Resource tags** are another way to organize resources. Tags provide extra information, or metadata, about your resources. This metadata is useful for:

1.**Resource management 2.Cost management and optimization 3.Operations management 4.Security 5.Governance and regulatory compliance** 6.**Workload optimization and automation**

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. 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).

**A resource tag** consists of **a name** and **a value**. You can assign one or more tags to each Azure resource. (you might decide that only mission-critical resources have the Impact tag. All non-tagged resources would then not be considered as mission-critical.)

**Tags allow you to associate metadata with a resource to help keep track of resource management, costs and optimization, security, and so on.**

**The Total Cost of Ownership calculator lets you input your current infrastructure and requirements and provides you with an estimate for running in the cloud.**

# 3.2:Describe features and tools in Azure for governance and compliance

## Learning objectives

1. Describe the purpose of Microsoft Purview
2. Describe the purpose of Azure Policy
3. Describe the purpose of resource locks
4. Describe the purpose of the Service Trust portal

**Microsoft Purview** is a family of data governance, risk, and compliance solutions that helps you get a single, unified view into your data. Microsoft Purview brings insights about your on-premises, multicloud, and software-as-a-service data together.

With Microsoft Purview, you can stay up-to-date on your data landscape thanks to:

* Automated data discovery
* Sensitive data classification
* End-to-end data lineage

Two main solution areas comprise Microsoft Purview: **risk and compliance:**

* Protect sensitive data across clouds, apps, and devices.
* Identify data risks and manage regulatory compliance requirements.
* Get started with regulatory compliance.

and **unified data governance**:

* Create an up-to-date map of your entire data estate that includes data classification and end-to-end lineage.
* Identify where sensitive data is stored in your estate.
* Create a secure environment for data consumers to find valuable data.
* Generate insights about how your data is stored and used.
* Manage access to the data in your estate securely and at scale.

**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.

Azure Policy enables you to define both individual policies and groups of related policies, known as initiatives. 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. 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(CI/CD)policies** that pertain to the pre-deployment and post-deployment phases of your applications.

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.

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

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.

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.

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.

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 (Microsoft Entra organization account)

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**.
* 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.

**Azure policy lets you create policies and initiatives (groups of policies) that prevent non-compliant resource from being created.**

**A resource lock can be used to prevent a resource from accidentally being deleted.**

# 3.3:Describe features and tools for managing and deploying Azure resources

## Learning objectives

1. Describe Azure portal
2. Describe Azure Cloud Shell, including Azure CLI and Azure PowerShell
3. Describe the purpose of Azure Arc
4. Describe Azure Resource Manager (ARM) and Azure ARM templates
5. Describe infrastructure as code (IaC)

Azure provides multiple tools for managing your environment, including the:

1. **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

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.

1. **Azure PowerShell:**

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.

# 

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).

**3.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.)

**4.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.

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. 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)

**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.

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.

**Infrastructure as code** is a concept where you manage your infrastructure as lines of code. **ARM templates** and **Bicep** are two examples of using infrastructure as code with the Azure Resource Manager to maintain your environment.

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.

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

**Declarative syntax**: 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**:  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.

**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.

**Bicep** is a language that uses declarative syntax to deploy Azure resources. A Bicep file defines the infrastructure and configuration. Then, ARM deploys that environment based on your Bicep file. While similar to an ARM template, which is written in JSON, Bicep files tend to use a simpler, more concise style.

Some benefits of Bicep are:

**Support for all resource types and API versions**: As soon as a resource provider introduces new resource types and API versions, you can use them in your Bicep file. You don't have to wait for tools to be updated before using the new services.

**Simple syntax**: Bicep requires no previous knowledge of programming languages. Bicep syntax is declarative and specifies which resources and resource properties you want to deploy.

**Repeatable results:** Bicep files are **idempotent**, which means you can deploy the same file many times and get the same resource types in the same state. You can develop one file that represents the desired state, rather than developing lots of separate files to represent updates.

**Orchestration**:  Resource Manager orchestrates the deployment of interdependent resources so they're created in the correct order. When possible, Resource Manager deploys resources in parallel so your deployments finish faster than serial deployments. You deploy the file through one command, rather than through multiple imperative commands.

**Modularity**: Modules enable you to reuse code and simplify development

**Azure Arc, working with Azure Resource Manager, lets you extend your Azure compliance and monitoring to your hybrid and multicloud configurations.**

**Bicep and ARM Templates allow you to deploy your resource(Infrastructure) as code.**

# 3.4:Describe monitoring tools in Azure

## Learning objectives:

* Describe the purpose of Azure Advisor
* Describe Azure Service Health
* Describe Azure Monitor, including Azure Log Analytics, Azure Monitor Alerts, and Application Insights

**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.

# 

**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:

1. **Azure Status:**  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.
2. **Service Health** provides a narrower view of Azure services and regions. It focuses on the Azure services and regions you're using. 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.
3. **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.

**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. You can view high-level reports on the Azure Monitor Dashboard or create custom views by using Power BI and Kusto queries.

**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.

**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. 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%.

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**, an Azure Monitor feature, monitors your **web 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.

**The five recommendation categories for Azure Advisor are: Reliability, Security, Performance, Operational Excellence, and Cost.**

**Resource Health is a tailored view of your actual Azure resources. It provides information about the health of your individual cloud resources(EX:your VM).**

**Practise the dumps for confidence. ALL THE BEST! This is the short material which covers all the topics for AZ900 exam. Hope the highlighted notes help you~**