



Azure strategy and implementation guide for nonprofits

PUBLISHED BY
Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Copyright © 2018 by Microsoft Corporation

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

This book is provided “as-is” and expresses the author’s views and opinions. The views, opinions and information expressed in this book, including URL and other Internet website references, may change without notice.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

Microsoft and the trademarks listed at <http://www.microsoft.com> on the “Trademarks” webpage are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

Authors: Joachim Hafner, Simon Schwingel, Tyler Ayers, and Rolf McLaughlin (MASUCH). Introduction by Britt Johnston.

Editorial Production: Dianne Russell, [Octal Publishing, Inc.](#)

Copyeditor: Bob Russell, Octal Publishing, Inc.

Contents

Introduction	iv
Chapter 1: Microsoft Azure governance	1
Cloud envisioning	2
Cloud readiness	3
Cloud readiness framework	3
Organizational cloud readiness	10
Chapter 2: Architecture	13
Security	16
Securing the modern enterprise	17
Identity versus network security perimeter	19
Data protection.....	22
Data classification	22
Threat management.....	25
Identity	28
Azure AD.....	28
Domain topology	32
Connectivity	35
Hybrid networking.....	35
Global network design	38
Network security	39
Remote access.....	42
Application design	42
Microservices versus monolithic applications	43
Cloud patterns.....	44
Chapter 3: Application development and operations	50
Business application development	51
Waterfall to Agile to DevOps	52
DevOps.....	54

Moving to DevOps	63
Application operations	64
Secure	65
Protect	66
Monitor	77
Deep monitoring services	80
Integrate with IT Service Management	85
Integrate with Security Information and Event Management systems	85
Managed services for standard and business applications	90
Offering managed services	91
Consuming managed services	92
Provisioning managed services	93
Metering consumption	93
Billing and price prediction	94
Chapter 4: Service management	95
Incident management	96
Azure status	98
Service health	98
IT Service Management integration	100
Security incidents	100
Microsoft support	102
Problem management	103
Change management	104
Azure platform change	105
Capacity management	107
Asset and configuration management	108
Update and patch management	109
Azure platform	109
Software as a service	110
Platform as a service	110
Infrastructure as a service	110
Service-level management	112
Chapter 5: Conclusion	114
About the authors	115

Introduction

The value of technology to nonprofit organizations is what it can do to help you accelerate your mission and do more good. Nonprofits are constantly facing the challenges of fewer resources, greater need, and growing cybersecurity threats.

The right technology can help organizations like yours empower employees; drive innovation; broaden engagement with donors, volunteers, and beneficiaries; and protect data and internal systems from cyberthreats. One barrier to gaining those benefits is that many nonprofits struggle with old systems and manual processes that don't support newer and more versatile, agile, and secure technology solutions.

At Microsoft, we believe technology is a powerful force for helping nonprofit organizations achieve your missions and thus benefit society as a whole. So we are committed to supporting mission-driven nonprofits around the world by offering technology solutions that are accessible, affordable, and relevant.

\$1 billion cloud resources donated to nonprofits

As part of this commitment, we offer donations and discounts for nonprofits on a variety of cloud solutions. In early 2016, Microsoft Philanthropies pledged to donate \$1 billion in cloud computing resources over three years to help 70,000 nonprofit organizations digitally scale their impact and serve the public good. We reached our original goal a year early, benefiting more than 90,000 nonprofits of all sizes, dedicated to all kinds of missions. We have since increased our goal to reach over 300,000 nonprofits by 2020.

To support those efforts, we have built a growing ecosystem of partners who know the challenges that nonprofits face. These partners focus specifically on providing relevant technology and services, including Microsoft Azure solutions, to help nonprofit organizations scale and drive greater impact around the world through the cloud.

Hundreds of cloud services on a single platform

Microsoft Azure is an ever-expanding collection of cloud services that can help your organization handle your business challenges more securely, efficiently, and affordably. These services are hosted in more than 100 globally distributed datacenters and are supported by one of the world's largest multi-terabit global networks. Enterprise-grade management capabilities are built into the platform and available through a single, easy-to-use portal.

Move at your own pace

Moving to Azure is not an all-or-nothing process. You can move a workload or system at a time or maintain a hybrid cloud environment where on-premises and cloud capabilities run seamlessly together across infrastructure, data, users, and applications. The following are just some of the ways you can use the cloud to accelerate your mission:

- Equip your staff with familiar productivity tools that enable strong collaboration, communication, and creativity.
- Integrate end-to-end data intelligence to gain deeper insights across on-premises and cloud infrastructure to better understand needs, improve programs, and increase impact.
- Extend your datacenter and free up resources to directly contribute to your organization's mission.
- Prevent your IT team from getting bogged down in managing infrastructure so they can focus on delivering higher-impact applications.
- Keep beneficiary and donor information safer yet easily accessible across multiple platforms, devices, and locations.
- Easily scale up your cloud presence for a specific project or event and scale down when completed for greater flexibility at an affordable price.

Nonprofit-specific guidance and support

To make the transition to the cloud even easier for nonprofits, [AzureFoundation for nonprofits](#) provides prescriptive scripts, documentation, and diagrams to accelerate adoption. It also offers governance and security guidance specific to the most common nonprofit scenarios.

In deciding how, when, and what to move to the cloud, you need to know what adjustments your IT organization will need to make to start the transition. That's where this guide comes in. It can help decision makers gain a high-level overview of cloud-related topics, and it provides your IT team with the information they need to effectively implement and integrate cloud technologies and manage a hybrid environment. It covers key cloud concepts as well as Azure-specific tips and tactics related to:

- Recommended general architectures that incorporate security, identity, and cloud design principles
- DevOps and how it fits with cloud technologies
- Integrating, planning, and managing cloud resources
- Microsoft Azure governance

One of the key messages we hope you'll take away from this guide is that moving to the cloud does not mean that everything needs to change. Making adjustments and exploring new ideas in key areas can unlock a world of potential and make cloud transformation possible as well as productive for your organization.

Microsoft Azure governance

When it comes to governance a variety of interpretations exist. For Microsoft, Azure governance has three components (Figure 1-1): Design of Governance, Execution of Governance, and Review of Governance. In this chapter, we take a look at each of these components.

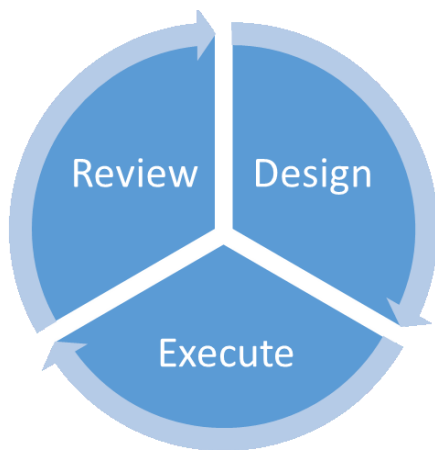


Figure 1-1: Azure governance

The first component is *Design of Governance*. This component derives from the customer's cloud vision. It comes together with the customer's constraints, such as regulatory obligations or privacy-related data that needs to be processed. This is the *why do we do things*.

The second component is *Execution of Governance*. This component contains all of the measures to fulfill the required needs, like reporting, encryption, policies, and so on, to ensure that the defined component is followed by measures that can be implemented and controlled. This is the *how we do things*.

Finally, to ensure that all of the measures are fulfilling the intended purpose, a *Review of Governance* is needed to verify that the implementation follows the design.

Cloud envisioning

Having digital transformation in mind is already the first step toward envisioning how you can use cloud technology. The next step is to break it down into actionable steps within your enterprise. For additional reading on the topic of cloud strategy and envisioning, we recommend reading *Enterprise Cloud Strategy*, 2nd Edition by Eduardo Kassner and Barry Briggs. You can find it at <https://azure.microsoft.com/resources/enterprise-cloud-strategy/>. The book explains and elaborates on actionable ideas, several of which are briefly introduced in the section that follows.

Figure 1-2 and the following list present the main pillars and how to understand them:

- **Engage your customers.** To deliver personalized, rich, connected experiences on journeys that your customers choose.
- **Transform your products.** To keep up with your fast-moving customers, efficiently collaborating to anticipate and meet their demands.
- **Empower your employees.** To increase the flow of information across your entire business operations, better manage your resources, and keep your business processes synchronized across all boundaries.
- **Optimize your operations.** To expand the reach of your business using digital channels, anticipate customer needs, understand how your products are used, and quickly develop and improve products and services.

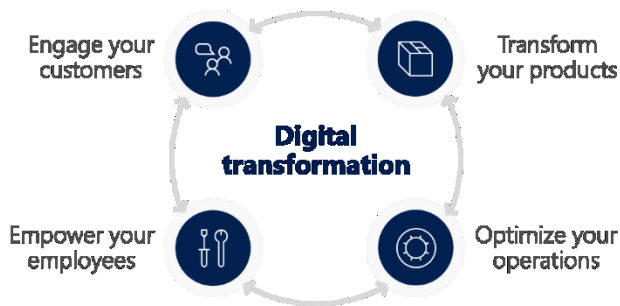


Figure 1-2: Digital transformation

Each of these pillars are connected to one another; some customers work on all pillars at the same time, whereas others are working on only one pillar at a time. This depends on the strategic decisions, capabilities, and capacities each customer can assign to the process and the defined timeline.

The top management task is now to name the action areas, give them priorities and the needed resources, and articulate the desired outcomes. This should be considered the company's North Star for orientation, for how to get there. Some enterprises prefer the "top-down" way of defining this, whereas others engage with their workforce for the same purpose.

Examples of cloud visions are, "We want to have 50 percent of our compute power moved to the cloud by 2020," or, "All our new products will be completely cloud-based on DevOps methodologies starting this fiscal year."

If there is a shared vision that guides the company as a whole through the digital transformation, the mission is accomplished.

Cloud readiness

After the cloud is envisioned as a means for the company's further evolution, the next steps need to be prepared and implemented. Here, envisioning and a clear picture can help you to keep track of your actions and let you prioritize to achieve quick wins while keeping the focus on the digital transformation. Cloud readiness is the next phase. But, to be certain, cloud readiness applies to more than a traditional waterfall project with its highly structured work breakdown structure (WBS). An Agile Scrum approach can be very successful, too, if the cloud vision and the desired outcome are well defined.

In the sections that follow, we describe areas we've identified in which change might be needed for an enterprise to handle cloud services effectively. The list is not exhaustive, and you should consider it as a starting point. If you identify additional areas in your organization that might need to undergo change, you are already in the driver's seat for your digital transformation.

Chapter 2 and Chapter 3 focus on developing a readiness framework and the organizational readiness to support a digital transformation. Following that, this e-book concentrates on more technical aspects, like Azure architecture, application development, and operations, we close with the service management of Azure and an outlook.

Cloud readiness framework

A *readiness framework* can help you to embed your cloud activities into your existing procedures, operational tasks, and responsibilities to make sure that you, as the enterprise, stay in control of your cloud journey. For some companies, the creation of a readiness framework is a huge task because their existing structures are challenged in a way that is very demanding. But that is the basic principle of the digital transformation.

Figure 1-3 is intended to serve as guidance for the next chapters and to give you a high-level overview of the areas that are relevant to Azure and its governance.

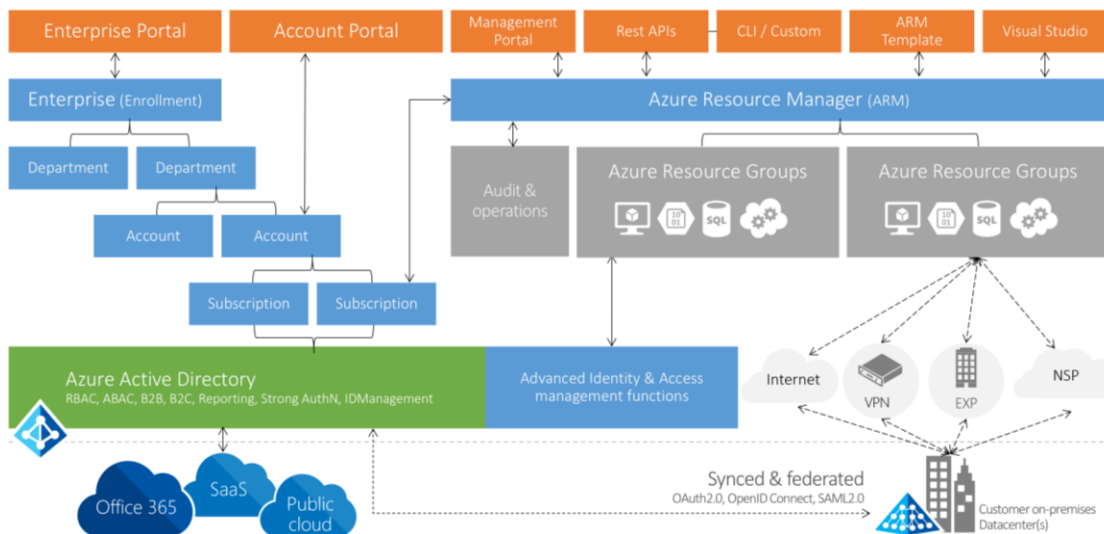


Figure 1-3: Cloud readiness framework

The orange blocks in the upper left, Enterprise Portal and Account Portal, with the components Enterprise, Department, Account, and Subscription below them show the dependencies from the enterprise contractual level down to the more technical element of Azure subscriptions. This part is mainly focusing on the contract, the purchase, and the billing of Azure. But already we can see that there is a strong dependency on Azure Active Directory (Azure AD).

Azure AD is the identity repository for other Microsoft Cloud services like Microsoft Office 365 and Microsoft Dynamics 365. Many enterprises choose to synchronize all or a major part of their on-premises Active Directory with Azure AD. Microsoft's recommended technology for this is the Azure Active Directory Connector, which is free of charge. In this way, companies remain in control of their corporate identities. A combination with federation services is possible and very often used as a means of stronger control.

With this level of control, the resources and different ways to interact with Azure are securely accessible through the common interface of the Azure Resource Management layer.

The Azure resource groups now are the main structure where all of your resources—for example, virtual machines (VMs), Azure storage, and platform services like Azure Machine Learning and so on—are grouped together.

Access to these resources is possible over the internet through secure channels like Virtual Private Networks (VPNs) or Azure ExpressRoute, as long as a dedicated Multiprotocol Label Switching (MPLS) connection with high-bandwidth options and Service-Level Agreements (SLAs) are in place.

Development operations model cloud services

One part of the framework is an operations model that is fit for the purpose of cloud services. The crucial point for many customers is the shift away from an oftentimes years-long, outsourcing model with a huge amount of infrastructure components to an Agile model with a blend of infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). To be clear, a cloud provider is worlds apart from an outsourcing provider, but we have seen customers that needed to evolve to the new way of interaction and defining the responsibilities to operate their new services in the cloud to be successful.

Customers that have successfully gone that way interacted early with their outsourcing providers and elaborated on new models of a managed cloud service that they could offer instead of the outsourcing model.

Chapter 3 provides more information regarding DevOps, operations, and other related topics.

Cost and order management

To achieve a level of cost transparency and to be able to assign certain cost alerts and limits, the adopting companies must integrate the Azure monetary model into their processes. We cannot totally describe the requirements here, and they can change in future versions of Azure. Sample requirements are Azure Usage and Azure Rate Cards. These are supported through the Azure Billing API and Azure Cost Management. Azure provides a special [billing API](#) that you can use to build your own solution for billing. Typical scenarios include the following:

- Azure spends during the month
- Set up alerts
- Predict bill
- Preconsumption cost analysis
- What-if analysis

Azure Cost Management—powered by [Cloudyn](#), which was [acquired by Microsoft](#) in July 2017—focuses on three main areas of the customer's cloud business. Table 1-1 lists these areas and their features.

Table 1-1: Azure Cost Management features

Focus area	Features
Gain real-time visibility to the Azure cloud environment	Keep track of upfront compute commitments and fees compared with actual consumption
	Reconcile prepay commitments with billing payments
	Verify Enterprise Agreement (EA) discounts with actual bills
	Stay on top of expiring resources and agreements
Empower enterprise-wide cloud accountability	Facilitate accurate cost allocation and chargeback across your enterprise entities including subscriptions, accounts, departments and cost centers
	Implement your own cost allocation method—blended/average/normalized rates, RI (Reserved Instance) autonomous rates, or any other policy of your choice
	Assure RI autonomy—assign zero costs to RI owners and add On-Demand costs to the departments that used external/borrowed RIs
	Track Azure Resource Manager groups' tags for simplified cost allocation
Drive Azure cost management and optimization	Monitor your VMs' performance-to-price ratio and receive actionable recommendations to maximize usage
	Calculate your most cost-effective upfront monetary and usage commitment
	Release unused Reserved IP addresses of stopped instances
	Dispose unattached block-blob storage volumes
	Apply changes directly through the Azure REST API

Another approach can be to use a third-party provider solution like [CloudCruiser](#).

Procurement (Provider to Azure customer)

Another area that will change with cloud adoption is the company's procurement. Ordering cloud services is very different from ordering boxes of software or buying blocks of licenses. It begins with adding licenses as needed; negotiating the EA; and understanding the subscription model of Azure, Office 365, and other cloud services from Microsoft.

Therefore, the procurement department needs to be a first-class citizen of the cloud-ready world. It begins by making the procurement team aware of the changes in the products and the way in which they are purchased. You must modify existing processes, which are based on buying boxed software and assigning it to cost centers, to purchasing and maintaining one or many Azure subscriptions with a possible dynamic cost value per month assigned to projects or cost centers. Especially the as-yet-determined amount of money that needs to be allocated to the project or cost center are the challenges for every customer.

Most enterprise customers are already acquainted with the contractual construct of a [Microsoft Enterprise Agreement \(EA\)](#). Additionally, Azure offers the [Pay-As-You-Go model](#). This model includes no commitment, and you pay only for the services that you actually consume. Payment is handled via credit card or debit card. Because this is not controllable from a technical perspective, most customers prohibit using company credit cards for cloud services. Also, expensing cloud services is very often prohibited by company internal regulations. Your Microsoft sales representative can help you to find the appropriate option for your situation.

Order (Internal customer of Azure customer)

Some customers purchase Microsoft Cloud services like Azure and Office 365 via a central group or department and then charge individual departments for services consumed. To carry out this type of purchasing, the customer needs to develop and implement the needed infrastructure and organizational processes. Technically, Azure supports this kind of cross-charging by optionally “tagging” Azure resources. These tags are visible on the monthly statement that is issued to the customer. The statement is also available as an Excel file.

Billing

Azure employs a subscription-based billing model. All subscriptions are bundled into the statement that comes with the enterprise enrollment. Depending on the technical implementation of solutions, enterprises can have only a few or hundreds of subscriptions. Either way, you should have in place a method of billing the costs associated with the Azure resources used to be able to maintain transparency and control in the billing process. Very often a change of the cost center model is considered to reflect the new cost quality.

In addition to the aforementioned resource tagging option, most customers rename the Azure subscription to include pertinent information such as the cost center. Others add the department name or the name of the project owner. Each of these solutions work if no changes occur, such as assigning new cost centers or assigning a new project owner. If that happens, you must change the respective subscription names.

Whether your company can use this kind of model also depends on the way you are using Azure and how many Azure subscriptions you can handle. There is no one-size-fits-all approach to the question of how many subscriptions a company should have. Some companies choose to have as few as possible while watching the [limits of a single subscription](#).

Other companies have decided to go for a more granular approach and assign at least one subscription to each project they start. Both solutions as well as the myriad variations have their limits. In the limited-subscription model, you might reach the upper limit of a resource type and need to expand to more subscriptions. However, purchasing many subscriptions becomes a complex billing problem and can be the issue of connecting resources that reside within the subscriptions. Depending on the customer’s products, you must define technical and security models.

Security standards and policies

When transitioning to the cloud, security plays a crucial role for all enterprises, and Microsoft is constantly working on the products to adapt to the latest developments and support our customers’ security requirements where possible.

A good starting point for security and Azure is the [Trust Center](#). Here, customers can take advantage of a collection of resources that are specific to the topic. Customers in highly regulated industries like healthcare, or government entities, need to verify that the services of Azure comply with applicable security controls.

The Cloud Security Alliance (CSA) has prepared the [Cloud Control Matrix](#). You can use this matrix to assess the security risk of any cloud provider. As are many other reputable cloud providers, [Microsoft is a member of the CSA](#).

The special landing page for [Azure security](#) provides you with further details about the available security measures that you can use to protect your company. This page also offers further guidance with respect to all Azure-related security topics.

When it comes to Azure governance and security, several options exist, and you must always perform a balancing act between customer usability and security. Beyond the high-level options mentioned earlier, per-subscription customers can avail themselves of the built-in Security Center or define their

own compliancy rules on a more granular level by using [Azure policies](#). The duty for the governance now is to define the framework, which, from then on, cloud solution architects can use to protect the solutions that reside in Azure. For the latest recommendations on securing your Azure environment, visit the Azure Trust Center.

Rights and role model

Most companies established access rights and user roles for their IT services to support the business and fulfill regulatory requirements as well as to reflect organizational responsibilities and duties. An outsourcing contract is also very often a driver for rights and roles models.

To be able to maintain a clear pattern of duty and responsibility and have that reflected in the rights and role implementation, an adoption of the cloud permission models is highly advised.

With Azure, you can adopt a [Role-Based Access Control \(RBAC\) model](#). In addition to the long list of [built-in roles](#), you can create your own [custom roles](#).

The best practice is to make use of the built-in roles as much as possible; only if none of these roles apply should you create a custom role. Sometimes roles and duties are mixed. These are technical roles that you can assign to users or groups. Assigning multiple roles to a given user makes it possible for that user to carry out his duties. We know of cases in which customers invested a lot of effort in creating a custom role like User/Group/Computer-Operator, but it would have been easier to assign several existing roles to the users for the same purpose.

Tenant and subscription management

As part of the rights and role model, you must incorporate a new component: the Azure tenant and subscription model. It should reflect your design considerations for Azure tenancy. To help you understand better, the following is Microsoft's description of a tenant:

"In the cloud-enabled workplace, a tenant can be defined as a client or organization that owns and manages a specific instance of that cloud service. With the identity platform provided by Microsoft Azure, a tenant is simply a dedicated instance of Azure Active Directory that your organization receives and owns when it signs up for a Microsoft cloud service such as Azure or Office 365."

More info You can learn more by going to <https://docs.microsoft.com/azure/active-directory/active-directory-what-is>.

Each Azure tenant can have multiple Azure subscriptions assigned to it, and you should manage these following a dedicated rights and role model. When it comes to deciding whether to have one or multiple tenants, you need to consider several aspects, some technical as well as organizational:

- Authentication and directory—user management and synchronization
- Azure Service integration
- IaaS, PaaS, SaaS; for example, Citrix, WebEx, Salesforce (approximately 3,400 applications)
- Service workloads (Microsoft Exchange, SharePoint, Skype for Business, etc.)
- Administration (of the tenants)
- Support and Helpdesk (also across tenants)
- Licensing and billing

As the number of tenants increase, the complexity of the solution grows significantly. We recommend that you limit the number to the absolute minimum. Allow new tenants only if your organization is prepared and they have a dedicated purpose that matches the effort. Some of the most common reasons we see for customers to consider multiple tenants are the following:

- The customer is a conglomerate of different companies, and each one needs to be a separated legal owner for Office 365 services.
- Different companies or business units belonging to the same parent company are autonomous from an IT perspective and make different choices, at different times.
- Different companies or business units belonging to the same parent company require completely segregated administration privileges.
- Global user distribution: concerns about performance in accessing Office 365 services out of one single region.
- Datacenter location, which can be driven by regulatory requirements.

More info To learn more about subscription and account management, you can go to <https://docs.microsoft.com/azure/virtual-machines/windows/infrastructure-subscription-accounts-guidelines>.

Azure administration

Typically, Azure administration is understood to mean using the Azure administrative portal to carry out the tasks related to the solution you are developing. However, when it comes to Azure governance, administration might require a more holistic approach. Depending on your duties and tasks, you might need to visit one of the Azure portals listed in Table 1-2.

Table 1-2: Azure administration

Portal	URL	Purpose
Azure Enterprise portal	https://ea.azure.com	Managing the EA and creating department/account levels
Azure portal	https://portal.azure.com	The standard portal to work with Azure
Azure Account portal	https://account.windowsazure.com	Managing the subscriptions that are assigned to a specific account level.
Office 365 portal	https://portal.office.com	Managing the Office 365 productivity suite with a direct link to manage Azure Active Directory

Which portal you should use will depend on your business needs.

License management

Usually, each Azure service is licensed as a pay-as-you-go model. The same is true for follow-on support of the service. Some services come with support included in the monthly fee; for others, you might need to pay an extra fee. Be sure to read the summary of every installation to get the latest information about the licensing and support model it uses. We recommend contacting your Microsoft sales representative, who might be able to save you license fees through a negotiated EA or through [the Azure Hybrid Benefit](#).

In Microsoft terminology, license management usually was counting the servers or workstations as well as the installed products. This approach was mostly quantitative and did not focus on the individual user. Still, when it comes to Azure, licenses are needed, but they mostly come with

subscriptions like Azure subscriptions or Office 365 subscriptions. Again, these are quantitative. When it comes to server products and operating systems, it's a different story. Some of them are automatically licensed when installed via the Azure portal. Still some need additional licenses, sometimes even from third-party providers. So, if you use third-party products, you will receive a separate bill from that software provider.

License assignment

Having acquired the licenses on the operating system or the server products can result in the need to allocate the license to the product. You can carry out allocation manually by entering license keys into products or by using tools from the vendor to ensure the proper usage of licenses—consult the relevant product documentation for [virtual machine licensing](#).

A very different method applies to the subscription-based model of, for instance, Office 365 or Azure AD. You can assign acquired licenses to your tenant and you can see them in the portal. Still, a per-user assignment of licenses is required. Here are some examples for [licensing yourself and your users in Azure Active Directory](#).

Your organization as well as your technical implementation must pay attention to the life cycle of a user and make sure the right licenses are allocated to the right user object. Very often this process is added to the Identity and Access Management (IAM) processes that are already in place.

Naming conventions

Very often, a significant amount of effort is put into creating naming conventions. If such conventions exist in your company an adoption of the resources within Azure is required.

Here are the main areas of definition:

- Subscriptions
- Resource groups
- Resources
 - Storage accounts
 - VNets
- Applications
- REST endpoints

Some of the conventions can become very granular, and a few resources like storage accounts come with their own restrictions because of internet regulations (RFCs).

For resources that are required to be unique across a region or even the entire Azure platform, a definition that is too strict can be too tight. You can use a naming element that serves as a break-free component—for example, a counter or a random string at the tail—to circumvent the constraint. For further details, refer to this discussion on [Azure naming conventions](#).

Keep in mind that sometimes you might use the command-line interface (CLI) or PowerShell to handle Azure resources. But also, you might sometimes use the Azure portal. Your naming convention should take into account that requirement and insert a distinct qualifier at the beginning of a name instead of putting it at the end where it might be truncated in dialogs.

Organizational cloud readiness

Next to the functional changes that your company might need to drive its digital transformation, there is another scope that requires attention: the organization. Although the first part can be very often described and written down and affects people indirectly, the organizational readiness is directly aimed at empowering your employees.

New technologies, new ways of interacting, or new ways of delivering service inside the company or to our customers are huge changes, especially for established organizations. This kind of disruptive approach, which very often is used to be more successful or copy competitors, sometimes puts an enormous strain on the workforce. Examples are AirBnB and the hotel industry, and Uber and the taxi industry. To avoid AirBnB taking away more and more customers, hotel companies needed to react. The same was true for cab operators.

Now, for organizations to deliver something similar to Uber or AirBnB and to make sure they also find a new home in the changed world, some companies even use change agents during the process.

The following sections describe approaches that have proven to be valuable for enterprises having the organizational scope in mind.

Cloud competence center

Many customers have found it helpful to establish a “Center of Knowledge” for Azure within their organization. The intention behind that decision is to create a pool of knowledge from which others in the organization can learn. This group can help define blueprints to ensure that all of the needed requirements such as security, operations, models, and so on are considered while the new, cloud-based solution is designed and implemented.

Definition of needed skills and training requirements

There are a lot of training programs available to help companies successfully make the transition to Azure and to build up a highly skilled team. Which program you choose very much depends on your business. Financial considerations are another aspect that you should consider.

There are many online courses as well as instructor-led courses available. Many of them are free of charge. Some are delivered through partners and certified trainers. Table 1-3 provides information about some of these courses.

Table 1-3: Azure training and certification

Topic	Resource
General skill training	https://azure.microsoft.com/training/
Microsoft Virtual Academy	https://mva.microsoft.com/search/SearchResults.aspx#!q=Azure
Azure certification	https://www.microsoft.com/learning/azure-exams.aspx
Design guidance, reference architectures, design patterns, and best practices	https://docs.microsoft.com/azure/#pivot=architecture
Azure Essentials	https://www.microsoft.com/azureessentials

Cloud readiness scopes

Following are typical areas of technology and knowledge:

- **Identity.** One of the cornerstones of the entire picture of Azure is the identity of a person. Microsoft sees the identity as the control plane of the modern world. You need to review your existing identity structure, which might be based on an on-premises implementation of Active

Directory, to determine whether it can serve the new purposes described earlier in this chapter, such as rights and role models or license assignment.

Many customers use this opportunity to review their current IAM system and modernize it to prepare it for new tasks. A profound knowledge about identities, the relation to Azure AD, and its security options should reside in the aforementioned competence center. Typical areas of interest include the following use cases:

- **Identity integration.** What has the company already in place to integrate identities into the application, and is it a more centralized approach or per-application with a relative independent model?

For the success of the future identity model, the company must decide which way to go for its identity integration. After the decision has been made, you can build a solution based on that decision. Plan for iterations in the design process to find the best solution and review against your security requirements.

- **Synchronization with on-premises Identity Management Systems.** What is the life cycle for identity management within the organization (in terms of user permissions and roles), and how can this process be securely extended for cloud identities?

Depending on the identity integration decision, you can develop a solution to ensure the required life cycle of user objects, group objects, and so on.

- **Authentication scenarios.** Will all solutions—whether on-premises or in the cloud—work with an identity that has been given to the user?

After you have designed the identity integration and identity the life cycle, we recommend a use case-based approach to ensure that all applications, on-premises and cloud based, can consume the identities the users will be assigned:

- **Multiforest considerations.** Does your company maintain a multiforest implementation on-premises due to historical or even regulatory reasons?

Some historical scenarios and some corporate decisions might not fit ideally in the cloud identity scenario. One example that we see very often is the corporate email address like contoso.com is a must, but all lines of business keep the requirement for their own mail servers, as well. You can do this in Azure with a combination of Office 365 and Azure AD, but this might not be the most effective solution. Also, if user objects for one individual are kept in several forests simultaneously, an easy way into the cloud is not possible. You need to consider some of the constraints of legacy implementations of Active Directory when it comes to connecting it with Azure AD.

- **Connectivity.** Another core component of all successful cloud adoptions is a very good knowledge about the network that is currently in place, how your applications are using the network, and how Azure is handling it.

There are multiple ways to securely connect your on-premises network to your private segment in Azure or give your next business application a secure home in any of your subscriptions, even without direct connectivity to your backbone. You can find a comprehensive list of options in Chapter 2.

As with identities, your competence center should be knowledgeable in all aspects of networking and how to connect to Azure.

- **Development.** Although the first two areas of knowledge are needed with every customer, the topic of development might not be needed for all. If you consider yourself as a company that needs cloud development to also be under governance control, the modern methods of cloud application development and terms like DevOps, or technologies like containers should be a priority. Chapter 4 presents more information for application development and operations recommendations.

Development of methodology for cloud integration

One of the major purposes of a cloud competence center in many enterprises is to define standards and develop methodologies for adoption of Azure services. This ensures a higher level of quality and makes sure that a good reuse of knowledge is achieved as well as a permanent alignment with the latest business or security requirements.

Development of cloud-integration blueprints: enterprise level

Some business solutions are designed for internal use only and have the individual contributor in the company as target for the implementation.

Based on the solution's requirements, the competence team should develop technology blueprints that ensure that the overall architecture of Azure with its tenant decisions, network recommendations, operational requirements, and so on are used and a proper SLA can be put on the final implementation.

Development of cloud-integration blueprints: partners, suppliers, and customers

Some other solutions are designed to cooperate with partners or are focused directly at the customer. Depending on the business of your enterprise, you might need additional blueprints to serve as a starting point for solutions that are accommodating that purpose.

You should pay special attention to the topic of identity, network, and security. You should expect these areas to become very complex depending on the scale of the solution or the integration needs of the solution.

Architecture

When an enterprise adopts Microsoft Azure as its cloud platform, it uses many different services that are owned and managed by multiple individuals. To allow for governance of its resources, Azure provides some general features, as illustrated in Figure 2-1.

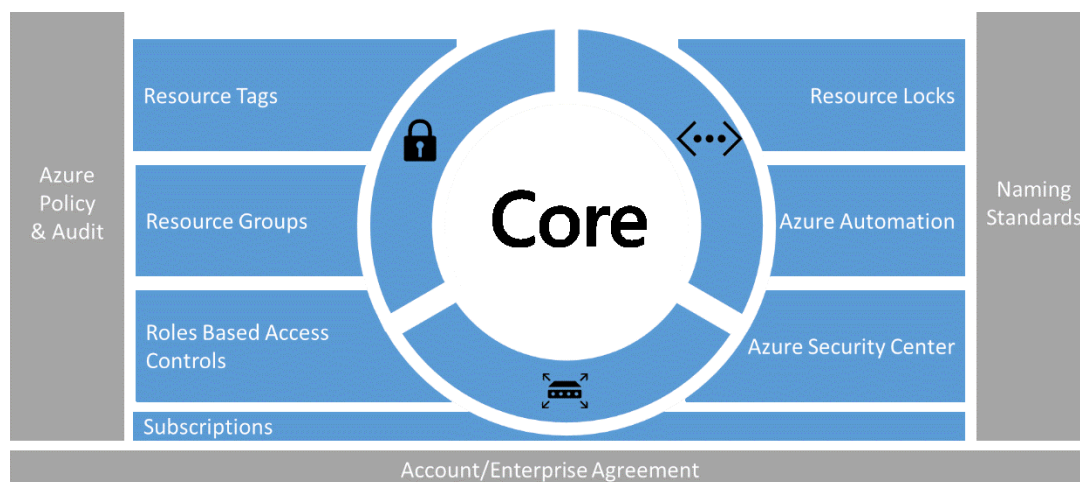


Figure 2-1: The building blocks of Azure governance

You should use *Naming Standards* to better identify resources in the portal, on a bill, and within scripts; for example:

- Follow the naming convention guidance (see the resources table in this section)
- Use camelCasing
- Consider using Azure Policies to enforce naming standards

You can use Azure Policies to establish conventions for resources in your organization. By defining conventions, you can control costs and more easily manage your resources. For example, you can specify that only certain types of virtual machines (VMs) are allowed, or you can require that all resources have a particular tag. Policies are inherited by all child resources. So, if a policy is applied to a resource group, it is applicable to all of the resources in that group. You can use Azure Policies at the subscription level to enforce the following:

- Geo-compliance/data sovereignty

- Cost management
- Default governance through required tags
- Prohibit public IP addresses where applicable

The *Azure Activity Log* is a log that provides insight into the write operations that were performed on resources in your subscription (previously known as “Audit Logs” or “Operational Logs”). Using the Activity Log, you can determine the “what, who, and when” for any write operations (PUT, POST, and DELETE) taken on the resources in your subscription. You can also understand the status of the operation and other relevant properties. You can integrate the Activity log into existing auditing solutions.

You apply tags to your Azure resources to logically organize them by categories. Each tag consists of a key and a value. Use resource tags to enrich your resources with metadata such as the following:

- Bill to
- Department (or business unit)
- Environment (production, stage, development)
- Tier (web tier, application tier)
- Application owner
- Project name
- Composite app (any resources tagged with the same value are considered part of the same application service)
- VM workload (identifies the primary application workload running in the VM—i.e., SQL Server, Jenkins)
- VM role (identifies the role the VM is delivering in the larger service—for example, database or build server)

A *resource group* is a container that holds related resources for an Azure solution. The resource group can include all of the resources for the solution or only those resources that you want to manage as a group. You decide how you want to allocate resources to resource groups based on what makes the most sense for your organization. Following are some considerations regarding resource groups:

- All of the resources in your group should share the same life cycle. You deploy, update, and delete them together. If one resource, such as a database server, needs to exist on a different deployment cycle, it should be in another resource group.
- Each resource can exist in only one resource group.
- You can add or remove a resource to a resource group at any time.
- You can move a resource from one resource group to another group. For more information, see [Move resources to new resource group or subscription](#).
- A resource group can contain resources that reside in different regions.
- You can use a resource group to scope access control for administrative actions.
- A resource can interact with resources in other resource groups. This interaction is common when the two resources are related but do not share the same life cycle (for example, web apps connecting to a database).

Azure Role-Based Access Control (RBAC) gives you fine-grained access management for Azure. Using RBAC, you can grant only the amount of access that users need to perform their jobs. You should use predefined RBAC roles where feasible, and only define custom roles where needed. You should follow the principle of granting the *least required privilege*.

As an administrator, you might need to lock a subscription, resource group, or resource to prevent other users in your organization from accidentally deleting or modifying critical resources. You can set the lock level to `CanNotDelete` or `ReadOnly`. When you consider using *Azure resource locks* to protect resources from unintended deletion, assign Owner and User Access Administrator roles only to people who you want to allow removing locks.

Azure Automation provides a way for users to automate the manual, long-running, error-prone, and frequently repeated tasks that are commonly performed in a cloud and enterprise environment. You can use Azure Automation to define *runbooks* that can handle common tasks such as shutting down unused resources and creating resources in response to triggers. With Azure Automation, you can do the following:

- Create Azure Automation accounts and review community-provided runbooks in the gallery
- Import and customize runbooks for your own use

Azure Security Center helps you to prevent, detect, and respond to threats with increased visibility into and control over the security of your Azure resources. It provides integrated security monitoring and policy management across your Azure subscriptions, helps detect threats that might otherwise go unnoticed, and works with a broad ecosystem of security solutions. You can use the Security Center to keep control of the security status of resources in your subscriptions:

- Activate data collection of resources you want to be monitored by Security Center
- Consider activating advanced threat management to detect and respond to threats in your environment

In the next subsections, we do some deep dives into selected topics that you should consider when starting out with Azure as your public cloud platform. You also should refer to [Azure Onboarding Guide for IT Organizations](#), which can help customers that are new to Azure to get underway. It explains the basic concepts of Azure, whereas the guide you're reading now provides a blueprint and best practices to roll out Azure on a large scale for customers with solid knowledge and experience with Azure. Table 2-1 provides resources on Azure architecture.

Table 2-1: Architecture resources

Topic	Resource
Azure Enterprise scaffold	https://docs.microsoft.com/azure/azure-resource-manager/resource-manager-subscription-governance
Naming conventions	https://docs.microsoft.com/azure/guidance/guidance-naming-conventions
Azure policies	https://docs.microsoft.com/azure/azure-resource-manager/resource-manager-policy
Activity Log	https://docs.microsoft.com/azure/monitoring-and-diagnostics/monitoring-overview-activity-logs
Resource tags	https://docs.microsoft.com/azure/azure-resource-manager/resource-group-using-tags
Resource groups	https://docs.microsoft.com/azure/azure-resource-manager/resource-group-overview
RBAC	https://docs.microsoft.com/azure/active-directory/role-based-access-control-what-is

Azure locks	https://docs.microsoft.com/azure/azure-resource-manager/resource-group-lock-resources
Azure Automation	https://docs.microsoft.com/azure/automation/automation-intro
Security Center	https://docs.microsoft.com/azure/security-center/security-center-intro

Security

The first thing to understand about cloud security is that there are different scopes of responsibility, depending on what kind of services you are using. In Figure 2-2, green indicates areas of customer responsibility, whereas blue areas are the responsibility of Microsoft.

On-Premises	IaaS	PaaS	SaaS
	Applications		
	Data		
	Runtime		
	Middleware		
	O/S		
	Virtualization		
	Servers		
	Storage		
	Networking		

Figure 2-2: Scope of security responsibility

For example, if you are using VMs in Azure (infrastructure as a service, or IaaS), Microsoft is responsible for securing the physical network and storage as well as the virtualization platform, which includes patching of the virtualization hosts. But you will need to take care of securing your virtual network and public endpoints yourself as well as patching the guest operating system (OS) of your VMs.

This document focuses on customer responsibilities. Microsoft responsibilities are not within the scope of this document. Table 2-2 lists where you can find a whitepaper covering that topic.

Table 2-2: Security resources

Topic	Resource
Security responsibility	https://azure.microsoft.com/resources/videos/azure-security-101-whose-responsibility-is-that/
Microsoft measures to protect data in Azure	https://azure.microsoft.com/mediahandler/files/resourcefiles/d8e7430c-8f62-4bbb-9ca2-f2bc877b48bd/Azure%20Onboarding%20Guide%20for%20IT%20Organizations.pdf https://servicetrust.microsoft.com/
Azure Security	https://docs.microsoft.com/azure/security/azure-security
Microsoft Whitepaper Security Incident Response	http://aka.ms/SecurityResponsepaper

Securing the modern enterprise

Securing a modern enterprise is complex and challenging. Microsoft's cybersecurity portfolio can help you to do the following:

- Increase visibility, control, and responsiveness to threats
- Reduce security integration and vendor management costs

Following are the pillars of a secure modern enterprise (see Figure 2-3):

- **Identity.** Embraces identity as primary security perimeter and protects identity systems, admins, and credentials as top priorities
- **Apps and Data.** Aligns security investments with business priorities including identifying and securing communications, data, and applications
- **Infrastructure.** Operates on modern platform and uses cloud intelligence to detect and remediate both vulnerabilities and attacks
- **Devices.** Accesses assets from trusted devices with hardware security assurances, great user experience, and advanced threat detection

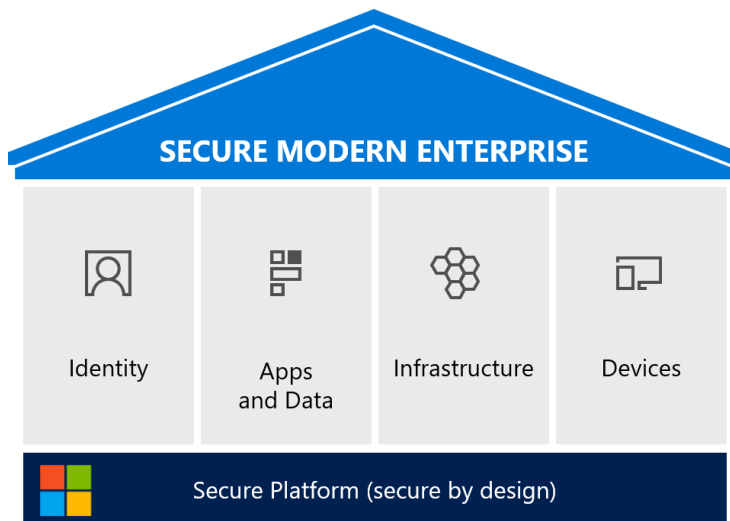


Figure 2-3: Components of a secure modern enterprise

Phase 1: Build the security foundation

The Securing Privileged Access (SPA) roadmap helps you to implement your security foundation (see Figure 2-4). This roadmap is structured in three stages. In stage 1, you mitigate the most frequently used attack techniques of credential theft by doing the following:

- Separating administrator accounts for administrator tasks
- Implementing Privileged Access Workstations (PAW) pattern for Active Directory administrators
- Using unique local administrator passwords for workstations
- Using unique local administrator passwords for servers

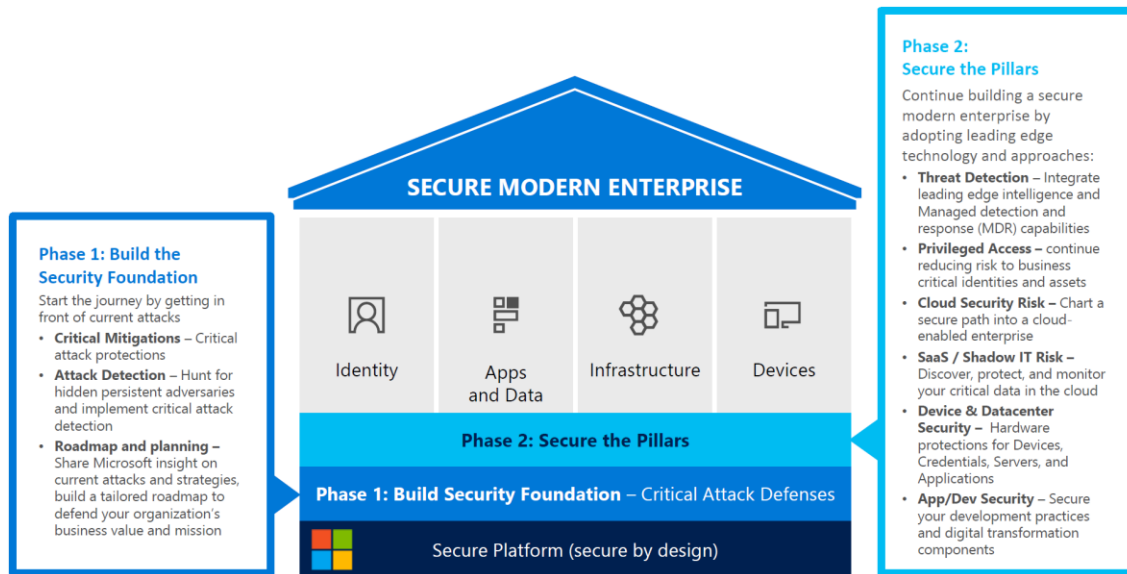


Figure 2-4: Build the security foundation phases

You can implement stage 2 of the roadmap in one to three months and build on the mitigations of stage 1:

- Follow the PAW pattern for all administrators and implement additional security hardening
- Time-bound privileged (Just in Time [JIT]) access for administrative tasks by using Azure Privileged Identity Management for Azure Active Directory (Azure AD), Privileged Access Management (PAM) for Active Directory, or JIT virtual machine access in Azure Security Center
- Enforce multifactor authentication (MFA) for privileged access
- Use the Just Enough Admin (JEA) pattern for domain controller maintenance
- Implement attack detection to gain visibility into credential theft and identity attacks by using Microsoft Advanced Threat Analytics (ATA), Azure Security Center advanced detection, and SQL Database threat detection

Stage 3 of the SPA roadmap strengthens and adds mitigations across the spectrum, allowing for a more proactive posture:

- Modernize roles and delegation model founded on JIT and JEA
- Smartcard or passport authentication for all administrators
- Administration forest for Active Directory administrators (ESAE)
- Code integrity policy for datacenters allows only authorized executables to run on a machine
- Shielded VMs for virtual datacenters prevent unauthorized inspection or theft by storage and network admins

Phase 2: Secure the pillars

The next step after covering the basics of your security foundation is to adopt additional, leading-edge security technologies, including threat detection and better intelligence:

- Build the strongest protections for *identity* systems and administrators. Protect identities with threat intelligence and hardware assurances.

- Increase visibility and protection for apps and data in the cloud and on-premises. Reduce the risk to the applications you develop, Software as a service (SaaS) apps, and legacy/on-premises apps.
- Integrate cloud infrastructure securely. Protect by using hardware integrity and isolation, detect with analysts and threat intelligence.
- Deploy hardware protections for *devices*, data, applications, and credentials. Use advanced attack detection and remediation technology and analyst support.

Table 2-3 lists the needed security resources.

Table 2-3: Security resources

Topic	Resource
Cybersecurity reference architecture	https://mva.microsoft.com/training-courses/cybersecurity-reference-architecture-17632
SPA roadmap	http://aka.ms/SPARoadmap
ESAE	http://aka.ms/esae
PAW	https://docs.microsoft.com/windows-server/identity/securing-privileged-access/privileged-access-workstations

Identity versus network security perimeter

It's a universal concept: When you want to protect something, build a perimeter around it. Traditionally, in IT this perimeter is at the network level in the form of a firewall. The purpose of a network perimeter is to repel and detect classic attacks, but attackers can reliably defeat them via phishing and credential theft. At the same time, your data is moving out of the organization via approved or unapproved cloud services. Last but not least, employees need to keep productive wherever they are, using whatever device they carry with them, meaning that your data might be accessed by unmanaged devices. Matching these new challenges requires you to build a new kind of perimeter in addition to your existing network perimeter: an *identity security perimeter*.

Why you need an identity security perimeter

Identity controls access to your data. To protect your data, protect it directly at the front door by implementing multifactor authentication and conditional access. Conditions for access might depend on the user's current location (or IP address), on the user's device state, or the user's general risk score. If any of these configurable conditions are met, a modern identity system either challenges the user for a second authentication factor or denies access entirely.

What challenges must be matched

Table 2-4 lists what strategies are used to overcome specific challenges and which Microsoft solution you can employ to implement a specific strategy.

Table 2-4: Cloud security challenge matrix

Challenges	Strategy	Solution
Phishing reliably gains foothold in environment Credential theft allows traversal within environment	Time-of-click (versus time-of-send) protection and attachment detonation Integrated intelligence, reporting, policy enforcement Securing privileged access roadmap to protect Active Directory and existing infrastructure	Microsoft Office 365 Advanced Threat Protection Azure Active Directory Identity Protection, Conditional Access Advanced Threat Analytics
Shadow IT SaaS management	Discover SaaS usage Investigate current risk posture Take control to enforce policy on SaaS tenants and data	Cloud App Security
Limited visibility and control of sensitive data Data classification is large and challenging	Protect data anywhere it goes Bring or hold your own key Support most popular formats Integration with existing Data Loss Prevention (DLP) systems	Azure Information Protection and Azure Rights Management Edge DLP Endpoint DLP
Provide secure PCs and devices for sensitive data Manage and protect data on noncorporate devices	Provide a great user experience, hardware-based security, and advanced detection + response capabilities Mobile Device Management (MDM) and Mobile App Management (MAM) of popular devices via Intune Policy enforcement via Conditional Access	Windows 10 Intune MDM/MAM, Conditional Access System Center Configuration Manager + Intune
Protect broad attack surface that includes applications, infrastructure, management and administrative practices	Protect Privileged Identities: <ul style="list-style-type: none"> Achieve least privilege with JEA and JIT Protect credentials in use with Credential Guard and Remote Credential Guard Harden Private Cloud Fabric <ul style="list-style-type: none"> Restrict what applications can run on servers (Device Guard) Protect applications with exploit mitigations (Control Flow Guard) and antimalware (Windows Defender) 	Active Directory MIM PAM Implement PAW Pattern Credential Guard Controls Flow Guard Windows Defender

Challenges	Strategy	Solution
	Secure workloads in the cloud and on-premises <ul style="list-style-type: none"> Minimize attack surface (Nano) and VM grade isolation (Hyper-V containers) Shielded VMs provide isolation and integrity for sensitive workloads 	
Increased complexity by adding cloud datacenters to existing on-premises infrastructure Limited IT security knowledge and tooling to secure cloud/hybrid infrastructure	Broad and deep visibility—get insights into security of your hybrid cloud infrastructure and workloads Provide familiar capabilities—via Security Information and Event Management (SIEM) integration and security capabilities in Azure Marketplace Start with high security—Secure platform, rapidly find and fix basic security issues, critical capabilities	Log Analytics Security Center, Threat Protection, and Threat Detection Security Appliances from Azure Marketplace DDoS attack mitigation, Backup and Site Recovery, Network Security Groups, Disk and Storage Encryption, Azure Key Vault, Azure Antimalware, SQL Encryption, Firewall, and Secure DevOps Kit for Azure, and Application Insights

A Microsoft cybersecurity architect can provide expert advice and help you to build your security roadmap and support the successful integration into your organization. Table 2-5 lists the available resources.

Table 2-5: Identity security perimeter resources

Topic	Resource
Microsoft Data Guard	https://www.microsoft.com/store/p/data-guard/9nblggh0j1ps
Microsoft Control Flow	https://msdn.microsoft.com/library/windows/desktop/mt637065(v=vs.85).aspx
Shielded VMs	https://docs.microsoft.com/system-center/vmm/guarded-deploy-vm
Azure Site Recovery	https://docs.microsoft.com/azure/site-recovery/
Azure Backup	https://docs.microsoft.com/azure/backup/
Key Vault	https://docs.microsoft.com/azure/key-vault/key-vault-what-is
Azure Antimalware	https://docs.microsoft.com/azure/security/azure-security-antimalware
Security Center	https://docs.microsoft.com/azure/security-center/security-center-intro
Privileged Identity Management	https://blogs.technet.microsoft.com/tangent_thoughts/2016/10/26/azure-pim-initial-walkthrough-and-links-aka-msazurepim/
DLP	https://blogs.msdn.microsoft.com/mvpawardprogram/2016/01/13/data-loss-prevention-dlp-in-sharepoint-2016-and-sharepoint-online/
Office 365 Advanced Threat Protection	https://blogs.microsoft.com/firehose/2015/04/08/introducing-exchange-online-advanced-threat-protection/#sm.00011o28ibuaefxph31q9rbgj49j
System Center Configuration Manager	https://www.microsoft.com/cloud-platform/system-center-configuration-manager
Intune MDM	https://docs.microsoft.com/intune/device-lifecycle

Topic	Resource
Intune MAM	https://blogs.technet.microsoft.com/cbernier/2016/01/05/microsoft-intune-mobile-application-management-mam-standalone/
Azure AD Conditional Access	https://docs.microsoft.com/azure/active-directory/active-directory-conditional-access
Windows 10 Security	https://docs.microsoft.com/windows/threat-protection/overview-of-threat-mitigations-in-windows-10
Application Insights	https://docs.microsoft.com/azure/application-insights/app-insights-overview

Data protection

Data is one of the most valuable assets that companies have, and it is critical that this asset is protected against unauthorized access or hostage takers. Access is controlled by giving authenticated users the authorization to read, change, or delete data. Some data might be less critical than other data; that is, information might be publicly available or strictly confidential.

Data classification

Most companies already have a data classification policy in place. You will need to understand how using Azure as an application platform will affect this policy.

Table 2-6 presents examples of data classes that you might want to differentiate.

Table 2-6: Data classification example

Class	Example
Public	Announced corporate financial data
Low business impact	Age, gender, or ZIP code
Moderate business impact	Address, operating procedures
High business impact	Design and functional specifications

For each data protection class, your policy should define at least the following:

- To which data class a given policy applies
- The party responsible for data protection
- Precautions to protect data against any unauthorized or illegal access by internal or external parties
- How data should be stored and backed up
- How you ensure data is accurate and up to date
- How long data will be stored
- Under what circumstances you will disclose data and to whom
- How to keep individuals informed about data you hold
- Where data will be stored or transferred to; for instance, countries having adequate data protections laws
- What to do in case of lost, corrupted, or compromised data

Review your data protection policy regularly; for instance, once every two years to keep it up to date with new technologies and changes in jurisdiction.

Table 2-7 lists the different features that Azure provides for you to implement your data protection policies.

Table 2-7: Customer-configurable protection options in Azure

Topic	Resource
Volume-level encryption	https://docs.microsoft.com/azure/security/azure-security-disk-encryption
Key management	https://azure.microsoft.com/resources/videos/azurecon-2015-encryption-and-key-management-with-azure-key-vault/
SSL certificates	https://docs.microsoft.com/azure/app-service-web/web-sites-purchase-ssl-web-site
SQL Database encryption	https://docs.microsoft.com/sql/relational-databases/security/encryption/transparent-data-encryption-with-azure-sql-database
Azure Rights Management services	https://docs.microsoft.com/information-protection/understand-explore/what-is-azure-rms
Azure Information Protection	https://docs.microsoft.com/information-protection/understand-explore/what-is-information-protection
Azure Information Protection samples	https://github.com/Azure-Samples/Azure-Information-Protection-Samples

Microsoft Azure Information Protection is a very powerful solution to enforce the appropriate protection of documents and emails according to your data protection policies.

Using Azure Information Protection, documents are classified at creation-time (see Table 2-8). The document is encrypted from creation for the rest of its life cycle. When an authenticated user has the appropriate permissions according to the applied policy, the document is decrypted, and the user is able to open it—independent of the current location of the document.

Table 2-8: Data classification resources

Topic	Resource
Carnegie Mellon University Guidelines for Data Classification	http://www.cmu.edu/iso/governance/guidelines/data-classification.html
Microsoft Data Classification Wizard	https://www.microsoft.com/security/data/
Microsoft Data Classification Toolkit	http://www.microsoft.com/download/details.aspx?id=27123
Azure Rights Management	https://docs.microsoft.com/information-protection/understand-explore/what-is-azure-rms
Protecting data in Microsoft Azure	http://go.microsoft.com/fwlink/?LinkID=398382&clcid=0x409

Managing data location

When you create an Azure storage account, you define in which Azure datacenter region you want your storage to be located. By choosing one of the following storage redundancy options, you determine how often and to which location your data will be replicated:

- **Locally redundant storage (LRS).** This is replicated three times within a single region. When you write data, the write operations are performed synchronously across all three replicas.
- **Geo-redundant storage (GRS).** This is replicated three times within a single region and additionally synchronized asynchronously to the paired region. Table 2-9 contains a link to a list of region-pairs.
- **Read-access geo-redundant storage (RA-GRS).** This is the same as GRS, but you also have read access to data at the secondary region.

For some Azure services, you also have the option to choose a geo-replication location at the application level. Table 2-9 provides the links to read more.

Table 2-9: Data location management resources

Topic	Resource
Azure SQL Database	https://docs.microsoft.com/azure/sql-database/sql-database-geo-replication-portal
Azure Cosmos DB	https://docs.microsoft.com/azure/documentdb/documentdb-distribute-data-globally
Azure Service Bus	https://docs.microsoft.com/azure/service-bus-messaging/service-bus-outages-disasters
Region Pairs	https://docs.microsoft.com/azure/best-practices-availability-paired-regions

Note Ensure that you are storing data in regions according to applicable laws for different countries.

Encryption

If encryption at rest is a required step to be compliant with your enterprise's data protection policy, you should consider implementing the following:

- **Azure Disk Encryption.** This protects any data written to drives within your VM. Individuals having access to the storage where your drives reside will not be able to read any data within the drive.
- **SQL data encryption (aka Transparent Data Encryption [TDE]).** This protects against malicious activity by performing real-time encryption of the database and associated logs and backups. As of this writing, SQL Database does not support Key Vault integration. If this is a requirement, you might want to consider setting up an instance of Microsoft SQL Server as an IaaS VM.
- **Azure Storage Service Encryption (SSE).** This protects against malicious access on physical storage volumes by encrypting data before it is written using 256-bit Advanced Encryption Standard (AES) encryption.

Key management for encryption is an integral part of any data protection solution. You can use Key Vault to manage keys required for any type of encryption in a secure, configurable, auditable, and monitorable way. Optionally, you can use a hardware security module (HSM) with Key Vault.

To protect your data in transit, always use Secure Sockets Layer (SSL)/Transport Layer Security (TLS) whenever data is shipped to different locations. Where possible, you should use IPsec tunnels or a dedicated high-speed WAN link such as Azure ExpressRoute to protect all data in transit. Table 2-10 presents additional resources on encryption.

Table 2-10: Encryption resources

Topic	Resource
Encryption best practices	https://docs.microsoft.com/azure/security/azure-security-data-encryption-best-practices
Azure Disk Encryption	https://docs.microsoft.com/azure/security/azure-security-disk-encryption
SQL data encryption	https://docs.microsoft.com/sql/relational-databases/security/encryption/transparent-data-encryption-with-azure-sql-database
SSE	https://docs.microsoft.com/azure/storage/storage-service-encryption
Key Vault	https://docs.microsoft.com/azure/key-vault/key-vault-what-is
HSM support for Key Vault	https://docs.microsoft.com/azure/key-vault/key-vault-get-started#a-idhsm-if-you-want-to-use-a-hardware-security-module-hsm

Threat management

Threat management consists of three phases:

- Threat prevention includes measures to prevent attackers from penetrating your security perimeter.
- Threat detection assumes that an adversary already penetrated your security perimeter, and you need to identify any malicious activities before the attacker begins acting against you.
- When you identify a threat, any mitigating action is called *threat response*.

Follow these recommendations to implement your threat management strategy:

- Prevention
 - Implement access by following the PAW infrastructure pattern
 - Separate administrative resources into a dedicated Administrative Forest (ESAE)
- Detection
 - Advanced Threat Analytics (ATA)
 - Enterprise Thread Detection (ETD) Managed Detection and Response (MDR)
- Response
 - Incident Response

Figure 2-5 presents an overview of the Microsoft intelligence security graph.



Figure 2-5: Microsoft intelligence security graph

Azure offers built-in advanced threat detection functionality through services like Azure AD (refer to Chapter 3 for details) and Security Center. This collection of security services and capabilities provides a simple and fast way to understand what is happening within your Azure deployments.

Log Analytics is Microsoft's cloud-based IT management solution that helps you to gather and query log data at scale of your on-premises and cloud infrastructure. Based on this log data, Security Center or other Microsoft and community solutions provide valuable services. It's automatically integrated with Security Center and easy to integrate with System Center components such as System Center Operations Manager to extend your existing security management investments into the cloud. It is also possible to integrate Log Analytics with other solutions; for example, SIEM systems.

The Security Center Overview dashboard, shown in Figure 2-6, provides a comprehensive view into your organization's IT security posture with built-in search queries for notable issues that require your attention. The Security Center overview dashboard is the home screen for everything related to security in your Azure infrastructure. It provides high-level insight into the security state of your computers.

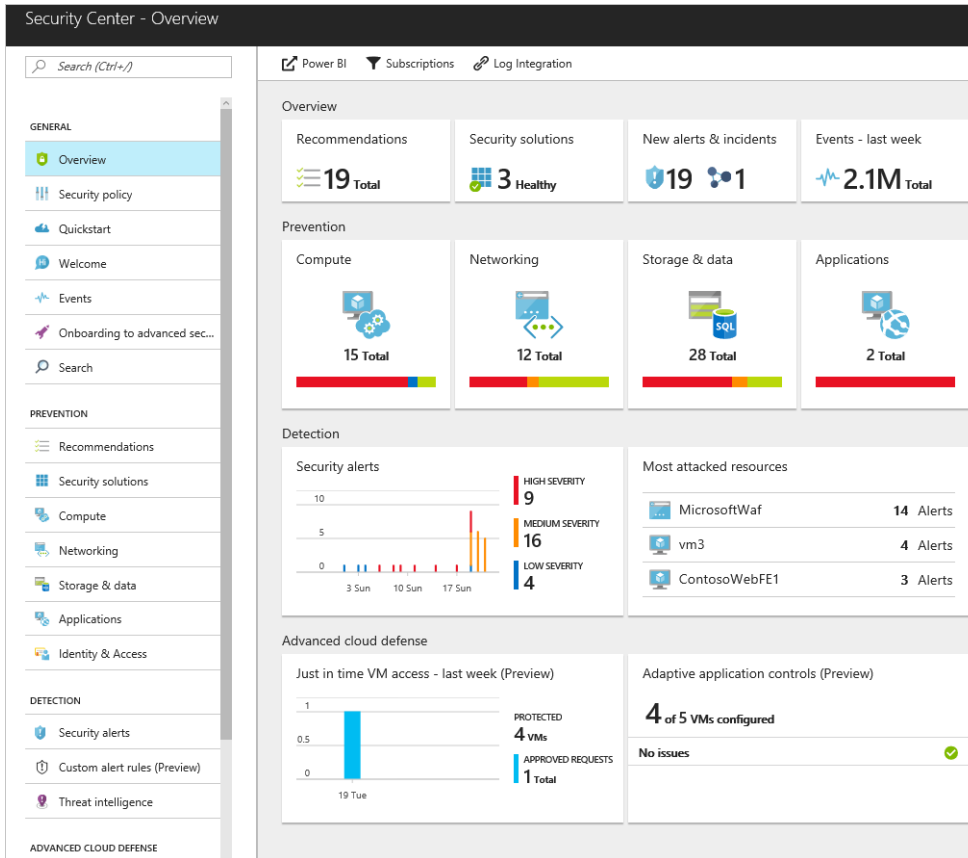


Figure 2-6: The Security Center Overview dashboard

Security Center helps you to quickly and easily understand the overall security posture of any environment, all within the context of IT operations, including software update assessment, antimalware assessment, and configuration baselines. Furthermore, security log data is readily accessible to streamline the security and compliance audit processes.

Security Center provides integrated security monitoring and policy management across your Azure subscriptions. Within the service, you are able to define policies against your Azure subscriptions and Resource Groups. Microsoft security researchers have access to an expansive set of telemetry gained from Microsoft’s global presence in the cloud and on-premises. This wide-reaching and diverse collection of datasets makes it possible to discover new attack patterns and trends. Thus, Security Center can rapidly update its detection algorithms as attackers release new and increasingly sophisticated exploits.

Security Center threat detection works by automatically collecting security information from your Azure resources, the network, and connected partner solutions. It analyzes this information, correlating information from multiple sources to identify threats. Security alerts are prioritized in Security Center along with recommendations on how to remediate the threat.

In addition to using Azure built-in services, you should establish the following professional services to detect and respond to threats (see also Table 2-11):

- **Enterprise Threat Detection.** This is a managed threat detection service that provides enterprises with state-of-the-art cyber-attack detection capabilities, taking advantage of the latest Microsoft cyber-defense technologies as well as security telemetry to help identify and mitigate potential threats.

- **Persistent Adversary Detection Service.** This is an engagement to proactively hunt for bad actors or malware that might be present in your environment using the same team, tools, technology, and telemetry that comprises our incident response (IR) offering (see Table 2-11 for details).
- **Investigation & Recovery.** With this service, Microsoft helps customers respond and recover from cyberattacks using its deep expertise on attacks, adversaries, malware, and Microsoft products. The service is effectively "on retainer" for customers with Premier support.

Table 2-11: Threat management resources

Topic	Resource
Azure Advanced Threat Detection	https://docs.microsoft.com/azure/security/azure-threat-detection
Log Analytics	https://docs.microsoft.com/azure/log-analytics/log-analytics-overview
Security Center	https://docs.microsoft.com/azure/security-center/security-center-intro
Enterprise Threat Detection	https://www.microsoft.com/security
Persistent Adversary Detection Service	http://download.microsoft.com/download/5/0/8/50856745-C5AE-451A-80DC-47A920B9D545/AFCEA_PADS_Datasheet.pdf
Investigation & Recovery	http://download.microsoft.com/download/5/1/6/516F59A7-91EE-4463-8612-C85FD3BEBDC7/microsoft-incident-response-and-recovery-process-brief.pdf
Incident Response offering	https://www.microsoft.com/microsoftservices/campaigns/cybersecurity-protection.aspx

Identity

As more and more of a company's digital resources reside outside the corporate network, in the cloud and on personal devices, a great cloud-based identity and access management solution is the best way to maintain control over, and visibility into, how and when users access corporate applications and data.

Azure AD

Azure AD is Microsoft's multitenant cloud-based directory and identity management service. All Microsoft online business services rely on Azure AD for sign-in and other identity needs. If you subscribe to any Microsoft online business, you get Azure AD with access to all free features. To enhance your Azure AD, you can add paid capabilities using the Azure Active Directory Basic, Premium P1, and Premium P2 editions. Here's what each one offers:

- Azure AD Basic is designed for task workers with cloud-first needs, this edition provides cloud-centric application access and self-service identity management solutions.
- Azure AD Premium P1 is designed to empower organizations with more demanding identity and access management needs, Azure AD Premium edition adds feature-rich enterprise-level identity management capabilities and makes it possible for hybrid users to seamlessly access on-premises and cloud capabilities.
- Azure AD Premium P2 is designed with advanced protection for all your users and administrators, this offering includes all of the capabilities in Azure AD Premium P1 as well as Identity Protection and Privileged Identity Management.

Table 2-12 summarizes the features offered by each Azure AD edition.

Table 2-12: [Azure AD feature overview](#)

Azure AD edition	Features
Free	<ul style="list-style-type: none"> • Directory objects limited to 500,000 • User/group management (add/update/delete)/user-based provisioning, Device registration • Single sign-on (SSO) • Connect (synchronization engine that extends on-premises directories to Azure AD) • Basic Security/usage reports
Basic	<ul style="list-style-type: none"> • Group-based access management/provisioning • Self-service password reset for cloud users • Company branding (sign-in pages/access panel customization) • Application proxy • Service-Level Agreement (SLA) 99.9 percent
Premium P1	<ul style="list-style-type: none"> • Self-service group and app management/self-service application additions/dynamic groups • Self-service password reset/change/inlock with on-premises write-back • Device objects two-way synchronization between on-premises directories and Azure AD (Device write-back) • Multifactor authentication (cloud and on-premises [MFA server]) • Microsoft Identity Manager user CAL • Cloud app discovery • Conditional access based on device state (allow access from managed/domain joined devices) • Automatic password rollover for group accounts
Premium P2	<ul style="list-style-type: none"> • Identity protection/Conditional access based on sign-in or user risk • Privileged Identity Management

When implementing your SPA roadmap, the features of Premium P2 are of great importance.

Azure AD Identity Protection is a feature of the Azure AD Premium P2 edition that provides you with an overview of the risk events and potential vulnerabilities affecting your organization’s identities. Identity Protection uses existing anomaly detection capabilities in Azure AD, which are available through its Anomalous Activity Reports.

Identity Protection uses adaptive machine learning algorithms to detect that an identity might have been compromised. Using this data, Identity Protection generates reports and alerts with which you can investigate and take appropriate mitigation action.

Based on risk events, Identity Protection (Figure 2-7) calculates a user risk level for each user so that you can configure risk-based policies to automatically protect the identities of your organization.

These risk-based policies, in addition to other conditional access controls provided by Azure AD and Enterprise Mobility + Security (EMS), can automatically block or offer adaptive remediation actions that include password resets and multifactor authentication enforcement.

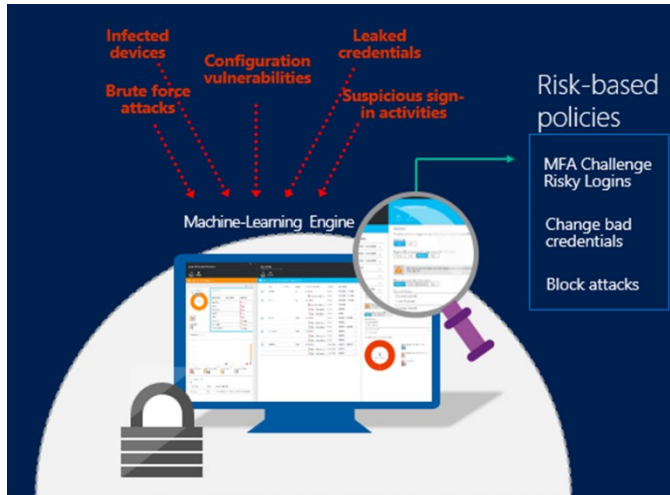


Figure 2-7: Azure AD Identity Protection

With Azure AD Privileged Identity Management, you can manage, control, and monitor access within your organization. This includes access to resources in Azure AD and other Microsoft online services like Office 365 or Microsoft Intune.

Azure AD Privileged Identity Management helps you to do the following:

- Get an alert and report on Azure AD administrators and JIT administrative access to Microsoft online services like Office 365 and Intune
- Get reports about administrator access history and changes in administrator assignments
- Get alerts about access to a privileged role

Hybrid identity is achieved by integrating your on-premises Active Directory with Azure AD using Azure AD Connect. You can use this to provide a common identity for your users for Office 365, Azure, and on-premises apps or SaaS applications integrated with Azure AD. With hybrid identity, you effectively extend your on-premises environment to the cloud for identity and access.

With Windows 10, Azure AD Enterprise State Roaming users gain the ability to securely synchronize their user settings and application settings data to the cloud to achieve the following:

- Separation of corporate and consumer data
- Enhanced security
- Better management

In addition to the features included in the different Azure AD editions, there are several Azure AD satellite services available, which we look at next.

Azure AD business-to-business

Azure AD business-to-business (B2B) collaboration capabilities make it possible for any organization using Azure AD to work safely and securely with users from any other organization, small or large. Azure AD B2B brings the following key features to you:

- Work with any user from any partner
 - Partners use their own credentials
 - No requirement for partners to use Azure AD
 - No external directories or complex setup required
- Simple and secure collaboration
 - Provide access to any corporate app or data, while applying sophisticated, Azure AD–powered authorization policies
 - Easy for users
 - Enterprise-grade security for apps and data
- No management overhead
 - No external account or password management
 - No sync or manual account life cycle management
 - No external administrative overhead

Azure AD business-to-cloud

Azure AD business-to-cloud (B2C) is a cloud identity management solution for your web and mobile applications. It is a highly available global service that scales to hundreds of millions of identities.

With Azure AD B2C, your application can authenticate to the following:

- Social accounts (such as Facebook, Google, LinkedIn, and more)
- Enterprise accounts (using open standard protocols, OpenID Connect, or Security Assertion Markup Language [SAML])
- Local accounts (email address and password, or username and password)

Azure AD Domain Services

Azure AD Domain Services provides managed domain services such as domain join, group policy, Lightweight Directory Access Protocol (LDAP), Kerberos/NTLM authentication that are fully compatible with Windows Server Active Directory. You can consume these domain services without the need to deploy, manage, and patch domain controllers in the cloud. Azure AD Domain Services integrates with your existing Azure AD tenant, thus making it possible for users to sign in using their corporate credentials. Additionally, you can use existing groups and user accounts to secure access to resources, thus ensuring a smoother “lift-and-shift” of on-premises resources to Azure Infrastructure Services. Azure AD Domain Services functionality works seamlessly regardless of whether your Azure AD tenant is cloud-only or synchronized with your on-premises Active Directory. Table 2-13 provides a list of security resources that you can use with Azure AD.

Table 2-13: Identity security resources

Topic	Resource
Azure AD overview	https://docs.microsoft.com/azure/active-directory/active-directory-what-is
Active Directory editions and features	https://docs.microsoft.com/azure/active-directory/active-directory-editions https://www.microsoft.com/cloud-platform/azure-active-directory-features
Azure Identity 101	https://docs.microsoft.com/azure/active-directory/understand-azure-identity-solutions
Azure AD Identity Management and Protection overview	https://youtu.be/9LGJ2-FKIM
Identity Protection	https://docs.microsoft.com/azure/active-directory/active-directory-identityprotection
Conditional access	https://docs.microsoft.com/azure/active-directory/active-directory-conditional-access-azure-portal
Privileged Identity Management	https://docs.microsoft.com/azure/active-directory/active-directory-privileged-identity-management-configure
Hybrid identity	https://docs.microsoft.com/azure/active-directory/connect/active-directory-aadconnect
Azure AD Domain Services	https://docs.microsoft.com/azure/active-directory-domain-services/active-directory-ds-overview
Azure AD B2B	https://docs.microsoft.com/azure/active-directory/active-directory-b2b-what-is-azure-ad-b2b
Azure AD B2C	https://docs.microsoft.com/azure/active-directory-b2c/active-directory-b2c-overview

Domain topology

Each Azure subscription is associated with an Azure AD tenant (see Figure 2-8). Azure AD manages identities on a tenant level. Those identities are used to grant access to resources for individuals. You can have multiple Azure AD tenants—for example, for production, QA, and development environments—but each Azure subscription can be associated with just one of those Azure AD tenants.

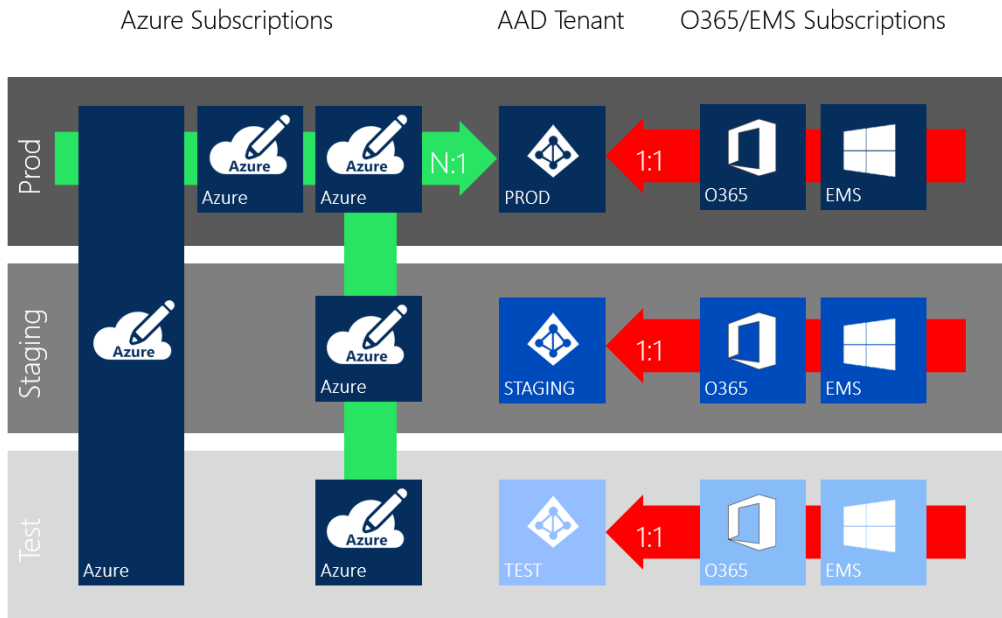


Table 2-8: Azure AD tenant-to-subscription relationship

Nevertheless, you can grant permissions on Azure resources to identities from other Azure AD tenants—either development tenants or tenants of business partners—by using Azure AD B2B features, as shown in Figure 2-9.

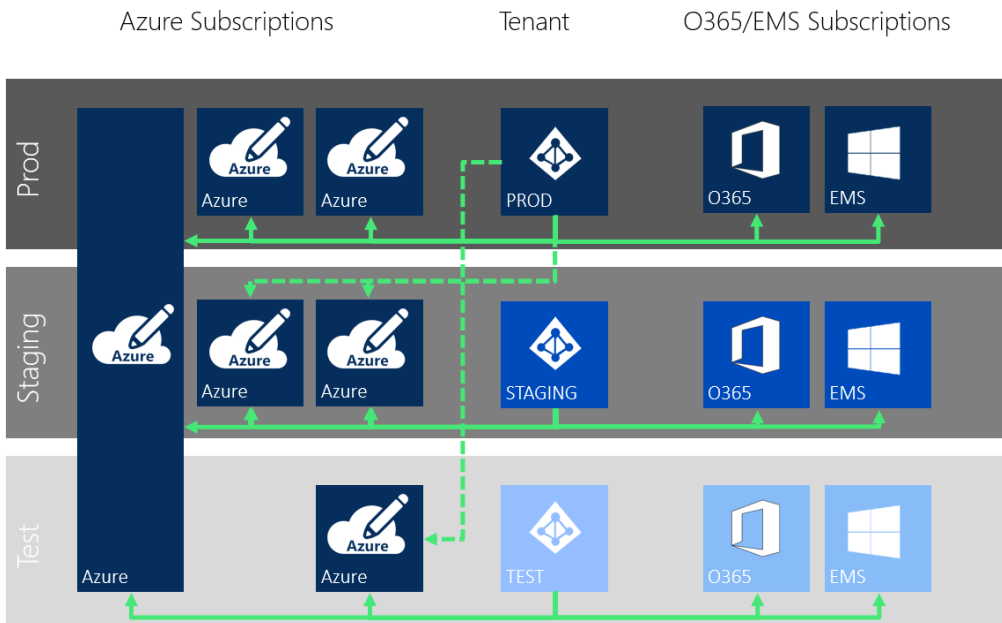


Table 2-9: Identities using Azure resources

Enterprise customers usually keep their on-premises Active Directory as the leading identity management system (see Figure 2-10). To use your existing identities, you need to synchronize all identities to Azure AD. SSO is supported by either synchronizing users' password hash between Active Directory and Azure AD by using Azure AD Connect or setting up Active Directory Federation Services for authentication or using pass-through authentication.

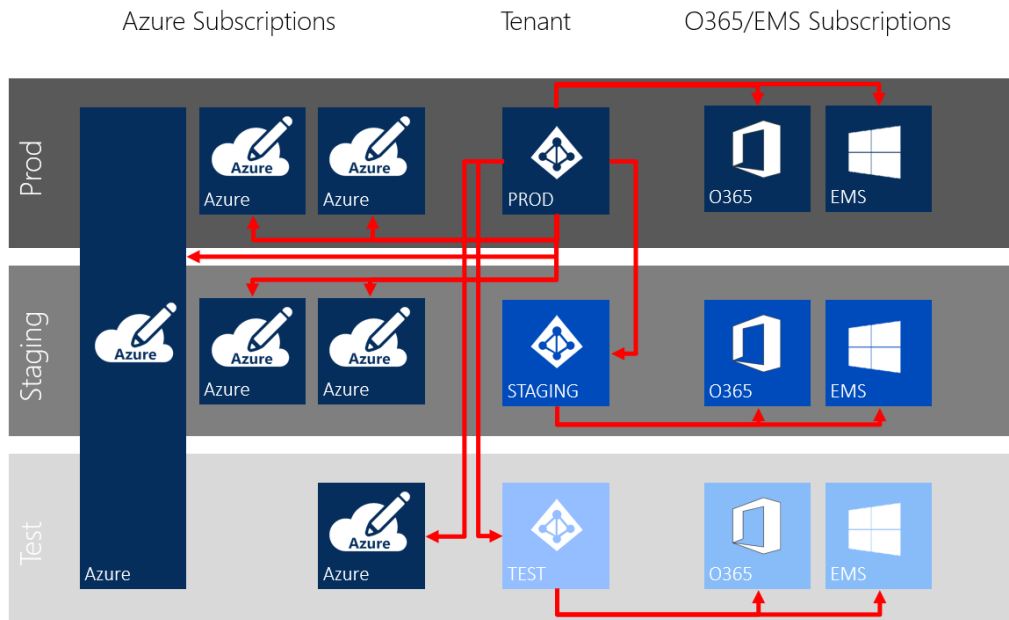


Figure 2-10: Identities managing subscriptions

You can join Azure VMs to a domain in two ways:

- Extend your on-premises Active Directory to your Azure virtual networks and join VMs to your existing on-premises domain
- Use Azure AD Domain Services

Most enterprise customers currently choose to extend their on-premises domain to Azure. Following best practices, you will implement at least a redundant pair of domain controllers in an Azure hub virtual network and configure a dedicated domain site for Azure for each Azure geo.

Table 2-14: Domain resources

Topic	Resource
Azure AD (including Azure AD Domain Services)	https://docs.microsoft.com/azure/architecture/reference-architectures/identity/azure-ad
Active Directory Connect	https://docs.microsoft.com/azure/active-directory/connect/active-directory-aadconnect
Federated authentication	https://docs.microsoft.com/azure/architecture/reference-architectures/identity/adfs
Pass-through authentication	https://docs.microsoft.com/azure/active-directory/connect/active-directory-aadconnect-pass-through-authentication
Azure AD seamless SSO	https://docs.microsoft.com/azure/active-directory/connect/active-directory-aadconnect-sso
Extending on-premises domain into Azure IaaS	https://docs.microsoft.com/azure/architecture/reference-architectures/identity/adds-extend-domain
Setting up datacenters on Azure	https://docs.microsoft.com/azure/active-directory/active-directory-deploying-ws-ad-guidelines
How Azure subscriptions are associated with Azure AD	https://docs.microsoft.com/azure/active-directory/active-directory-how-subscriptions-associated-directory
RBAC in Azure	https://docs.microsoft.com/azure/active-directory/role-based-access-control-what-is

Connectivity

Using the Azure Virtual Network service, you can securely connect Azure resources to one another by using virtual networks. A virtual network is a representation of your own network in the cloud. A virtual network is a logical isolation of the Azure cloud dedicated to your subscription. You also can connect virtual networks to your on-premises network.

Virtual networks are isolated from one another. You can create separate virtual networks for development, testing, and production that use the same Classless Inter-Domain Routing (CIDR) address blocks. Conversely, you can create multiple virtual networks that use different CIDR address blocks and connect networks together. You can segment a virtual network into multiple subnets. Azure provides internal name resolution for VMs and Cloud Services role instances connected to a virtual network. You can optionally configure a virtual network to use your own DNS servers instead of using Azure internal name resolution.

All Azure VM connected to a virtual network have access to the internet, by default. You also can turn on inbound access to specific resources, as needed.

You can connect Azure resources such as Cloud Services and VMs to the same virtual network. The resources can connect to one another using private IP addresses, even if they are in different subnets. Azure provides default routing between subnets, virtual networks, and on-premises networks, so you don't need to configure and manage routes.

You can connect virtual networks to one another; thus, resources connected to any virtual network can communicate with any resource on any other virtual network.

You can connect virtual networks to on-premises networks through private network connections between your network and Azure or through a site-to-site Virtual Private Network (VPN) connection over the internet.

You can filter inbound and outbound VM and Cloud Services role instances network by source IP address and port, destination IP address and port, and protocol.

You can optionally override Azure's default routing by configuring your own routes, or by using Border Gateway Protocol (BGP) routes through a network gateway. Table 2-15 provides a link to additional connectivity information.

Table 2-15: Connectivity resources

Topic	Resource
Azure Virtual Networks	https://docs.microsoft.com/azure/virtual-network/virtual-networks-overview

Hybrid networking

Connecting Azure virtual networks with your local network is referred to as *hybrid networking*. You can use the following concepts to access your virtual networks in Azure:

- **Internet connectivity.** All Azure resources connected to a virtual network have access to the internet, by default. You can also configure inbound access to specific resources, as needed, or turn off internet connectivity.
- **Site-to-site (S2S) VPN.** This connection uses an IPsec/IKE (IKEv1 or IKEv2) VPN tunnel. This type of connection requires a VPN device located on-premises that has a public IP address assigned to it and is not located behind a Network Address Translation (NAT). You can use S2S connections

for cross-premises and hybrid configurations. You can use either Azure VPN Gateway (a native Azure service) or deploy a third-party Network Virtual Appliance (NVA) to implement an S2S VPN.

- **Point-to-site (P2S) VPN.** Using this connection, you can create a secure connection to your virtual network from an individual client computer. P2S is a VPN connection over Secure Socket Tunneling Protocol (SSTP). P2S connections do not require a VPN device or a public-facing IP address to work. You establish the VPN connection by starting it from the client computer. This solution is useful when you want to connect to your virtual network from a remote location, such as from home or a conference, or when you have only a few clients that need to connect to a virtual network. You can use P2S connections with S2S connections through the same VPN gateway, as long as all the configuration requirements for both connections are compatible.
- **ExpressRoute.** This lets you extend your on-premises networks into Azure virtual networks over a dedicated private connection facilitated by a connectivity provider. Connectivity can be from an any-to-any (IP VPN) network, a point-to-point Ethernet network, or a virtual cross-connection through a connectivity provider at a colocation facility. ExpressRoute connections do not go over the public Internet. This makes it possible for ExpressRoute connections to offer more reliability, faster speeds, lower latencies, and higher security than typical connections over the internet

There is also the Service Bus Hybrid Relay option. This approach does not require a Layer 3 integration between your on-premises network and Azure; instead, it uses a secure socket connection via the internet.

To integrate your Azure-based resources with your on-premises resources, consider which network connectivity option fits best for you.

In Azure, ExpressRoute and VPN are both implemented physically redundant. If you need geo-redundancy, you will need to implement two separate ExpressRoute or VPN connections in different regions or even geo-locations.

For high availability with ExpressRoute, you should create ExpressRoute circuits with two different providers to provide resilience if one provider has a problem.

You can use the same concepts to integrate either Azure virtual networks to your on-premises networks or Azure virtual networks to Azure virtual networks—or Azure virtual networks to virtual networks in any other cloud. You can connect virtual networks with one another by using virtual network peering. Virtual network peering is generally available within a single region. (As of this writing, cross-regional peering is in public preview. As soon as it is generally available, Microsoft recommends using cross-regional peering instead of using an ExpressRoute circuit.) Many enterprise customers begin with an ExpressRoute circuit in one region.

Azure virtual networks in a region are typically organized in a hub-and-spoke network design pattern, as illustrated in Figure 2-11.

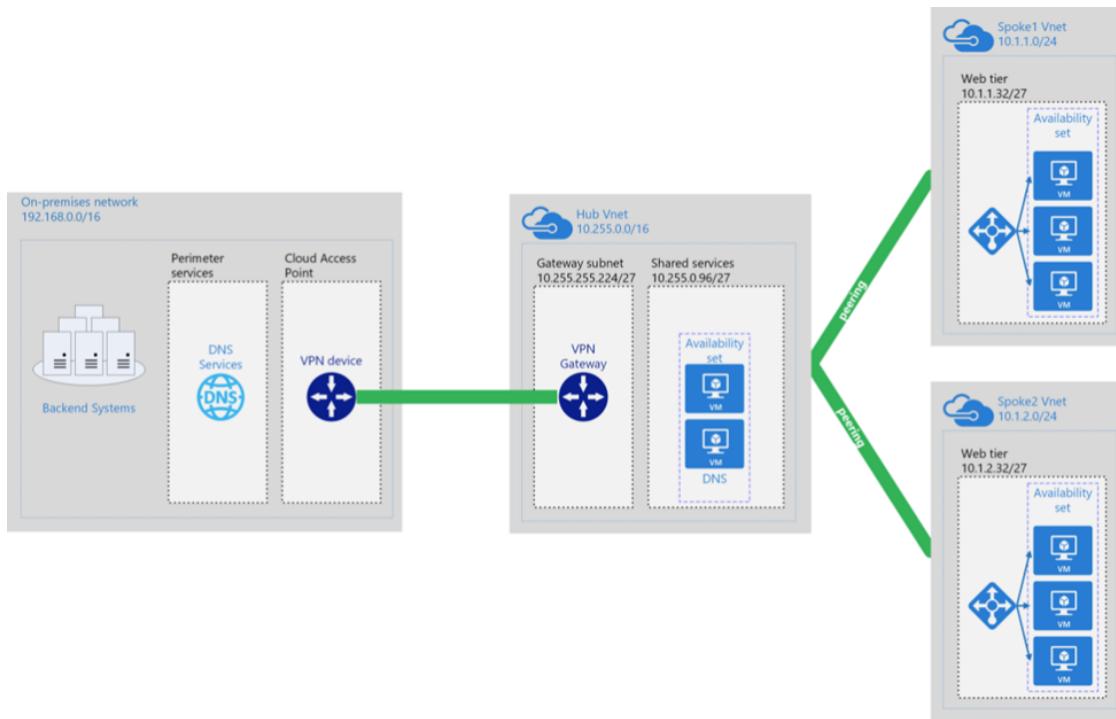


Figure 2-11: Hub-and-spoke network design sample

The hub virtual network contains the connection to the ExpressRoute circuit and any other centrally shared resources, like a next generation firewall and domain controllers for the new Azure site.

When setting up an Azure virtual network, you can configure whether to use Azure DNS service or your own DNS servers. Azure DNS automatically allows VMs in the same virtual network to resolve machine names to IP addresses as soon as a new VM is deployed. When using your own DNS server, you are responsible for creating appropriate DNS records for new VMs; that is, by joining them to your domain. Table 2-16 provides links to additional information.

Table 2-16: Hybrid networking resources

Topic	Resource
Azure networking basics	https://channel9.msdn.com/Blogs/Azure/Azure-Network-Security?ocid=player
Azure VPN gateways	https://docs.microsoft.com/azure/vpn-gateway/vpn-gateway-about-vpngateways
Integration options overview	https://docs.microsoft.com/azure/architecture/reference-architectures/hybrid-networking/considerations https://azure.microsoft.com/blog/networking-to-and-within-the-azure-cloud/?cdn=disable
VPN	https://docs.microsoft.com/azure/architecture/reference-architectures/hybrid-networking/vpn
ExpressRoute	https://docs.microsoft.com/azure/expressroute/expressroute-introduction https://docs.microsoft.com/azure/architecture/reference-architectures/hybrid-networking/expressroute
Intracloud integration options	https://azure.microsoft.com/blog/networking-to-and-within-the-azure-cloud-part-2/

Topic	Resource
Hub-and-spoke virtual network design	https://docs.microsoft.com/azure/architecture/reference-architectures/hybrid-networking/hub-spoke
High availability approaches	https://azure.microsoft.com/blog/networking-to-and-within-the-azure-cloud-part-3/
ExpressRoute best practices	https://docs.microsoft.com/azure/best-practices-network-security https://docs.microsoft.com/azure/expressroute/expressroute-optimize-routing https://docs.microsoft.com/azure/expressroute/expressroute-asymmetric-routing https://docs.microsoft.com/azure/expressroute/expressroute-routing-nat
Network security	https://docs.microsoft.com/azure/security/security-network-overview
Configure Routing	https://docs.microsoft.com/azure/virtual-network/virtual-networks-udr-overview
Quick-start template VPN	https://github.com/Azure/azure-quickstart-templates/tree/master/201-site-to-site-vpn
Azure DNS zone service	https://docs.microsoft.com/azure/dns/dns-overview
DNS for Azure VMs	https://docs.microsoft.com/azure/virtual-network/virtual-networks-name-resolution-for-vms-and-role-instances
Azure network security/DDOS	https://azure.microsoft.com/blog/azure-network-security
Azure Service Bus Relay	https://docs.microsoft.com/azure/service-bus-relay

Global network design

You can deploy a VM or any other resource only to a virtual network in the same subscription as the virtual network. Additionally, a virtual network can span only a single Azure region. If you need all of your resources to have a network connection to each other but need to separate your resources into different subscriptions and/or regions, you must use dedicated virtual networks in each subscription/region. To allow for network connectivity for resources in different virtual networks, it's possible to link virtual networks to one another. There are three options available to achieve this:

- Virtual network peering
- VNet2Vnet VPN
- ExpressRoute circuit

You can use virtual network peering to connect two Azure virtual networks, either within the same region or across Azure regions. VNet2VNet VPN connections are IPsec tunnels that are initiated between two virtual network gateways located within each virtual network. This can be either an Azure virtual network gateway or a virtual appliance of your choice. All traffic routed between different Azure regions will never leave the Microsoft networking backbone. Using one of these first two options makes it possible for you to link your virtual networks either in a mesh or a hub-and-spoke design.

If you use the third option, an ExpressRoute circuit, you will be able to use this circuit as a transfer net for your virtual networks as long as the virtual networks are in the same region as the ExpressRoute circuit. Traffic between your virtual networks is routed via your ExpressRoute circuit and will not leave the Microsoft networking backbone. If you want to connect a virtual network from another region or

geography, you must upgrade to ExpressRoute premium. Note that there are limits to how many virtual networks that you can connect to an ExpressRoute circuit. In this case (again), no network traffic leaves the Microsoft networking backbone. Figure 2-12 illustrates an example network topology of a customer with two subsidiaries connected to two Azure datacenter regions.

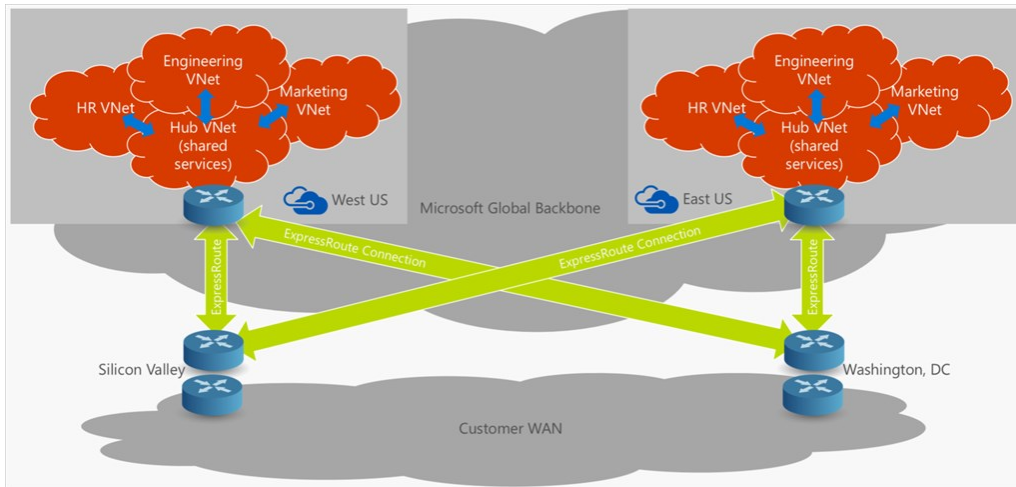


Figure 2-12: Sample network topology

Our customers' global network designs vary considerably. You can use any of the introduced building blocks to achieve the best individual solution, depending on your requirements. Table 2-17 provides links to global networking resources.

Table 2-17: Global networking resources

Topic	Resource
Virtual network peering	https://docs.microsoft.com/azure/virtual-network/virtual-network-peering-overview
VNet2VNet VPN	https://docs.microsoft.com/azure/vpn-gateway/vpn-gateway-howto-vnet-vnet-resource-manager-portal
ExpressRoute circuit	https://docs.microsoft.com/azure/expressroute/expressroute-howto-linkvnet-portal-resource-manager
ExpressRoute limits	https://docs.microsoft.com/azure/expressroute/expressroute-faqs#expressroute-premium
Design considerations	https://docs.microsoft.com/azure/virtual-network/virtual-network-vnet-plan-design-arm

Network security

Azure virtual networks are isolated from one another. Optionally, you can link virtual networks, as described in Chapter 3. Each virtual network supports the following default routing between devices:

- From within the same subnet
- From a subnet to another within the same virtual network
- From VMs to the internet

- From a virtual network to another virtual network through a VPN gateway
- From a virtual network to another virtual network through virtual network peering
- From a virtual network to your on-premises network through a VPN gateway

Enterprise customers usually have a security requirement forbidding any internet-bound traffic unless routed via a dedicated internet break-out that is secured by a next-generation firewall. You can address this requirement by implementing any or all of the following measures:

- **Network Security Groups (NSGs).** These contain a list of security rules that allow or deny network traffic depending on the source and destination address, port, and protocol. You can assign NSGs on either device or subnet level. For instance, you can block any internet-bound traffic by assigning an NSG with a “block any traffic to destination 0.0.0.0/0” rule.
- **User-Defined Routes (UDRs).** These specify the next hop for packets depending on the destination IP address. Using UDR, you can route internet-bound traffic either back to your on-premises network—hence, using your default internet break-out (aka forced tunneling)—or to an NVA. UDR tables are defined centrally in a subscription and can be assigned multiple times at subnet level.
- **NVA.** These are VMs based on a preconfigured image consisting of an operating system and a single application. In the Azure Marketplace, many NVAs are available for different purposes; for example, securing networking (firewall). This makes it possible for you to easily implement a perimeter virtual network containing NVAs for firewalling. Combined with a UDR that specifies this NVA as next hop for all internet-bound traffic, your NVA can filter and log all outbound traffic, as needed.
- **Next-Generation Firewall (NGFW).** This combines a traditional firewall with other network filtering functionalities such as application-level inspection among others. You will find NGFW solutions from Microsoft partners in the Microsoft Marketplace or you might opt for the Web Application Firewall (WAF) feature of Azure Application Gateway, which is provided as a platform-as-a-service (PaaS) offering.

To minimize potential threads, a general best practice for networking is separating internal- and external-facing resources in different virtual networks or subnets, as follows:

- Create virtual networks dedicated to external-facing workloads and internal-facing workloads.
- Configure network security groups to limit access. At a minimum, block access to the internet from internal virtual networks, and block access to the corporate network from external virtual networks.

Figure 2-13 illustrates an example topology for a perimeter network with a multilayered application in Azure.

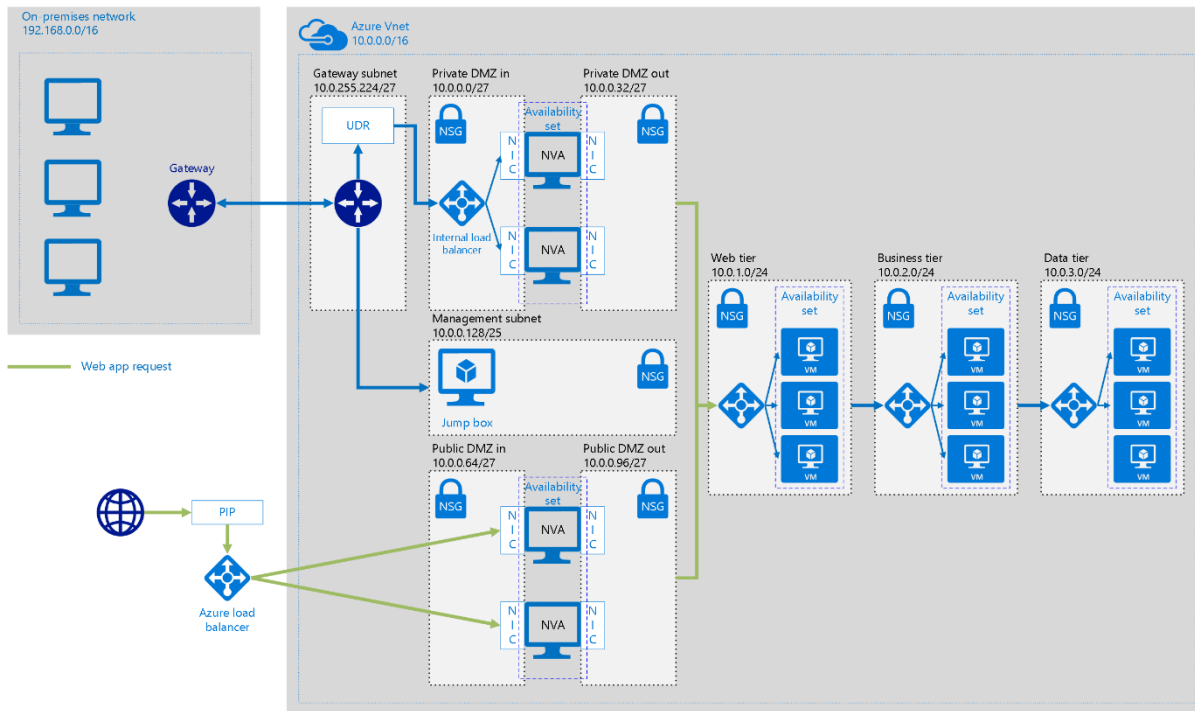


Figure 2-13: Sample perimeter network design in Azure

In Azure, monitoring is available on all network resources. Azure Network Watcher includes packet capture, next hop, IP flow verify, security group view, and NSG flow logs. Resource monitoring comprises diagnostic logs, metrics, troubleshooting, and resource health.

Table 2-18: Network security resources

Topic	Resource
Network security	https://docs.microsoft.com/azure/security/security-network-overview
Best practices for network security	https://docs.microsoft.com/azure/best-practices-network-security
NSG	https://docs.microsoft.com/azure/virtual-network/virtual-networks-nsg
NSG best practices	https://blogs.msdn.microsoft.com/igorpag/2016/05/14/azure-network-security-groups-nsg-best-practices-and-lessons-learned/
UDR	https://docs.microsoft.com/azure/virtual-network/virtual-networks-udr-overview
NVA	https://azure.microsoft.com/solutions/network-appliances/
NGFW	https://azuremarketplace.microsoft.com/marketplace/apps?search=next%20generation%20firewall&page=1
WAF	https://docs.microsoft.com/azure/application-gateway/application-gateway-web-application-firewall-overview
Sample DMZ designs	https://docs.microsoft.com/azure/architecture/reference-architectures/dmz/secure-vnet-hybrid https://docs.microsoft.com/azure/architecture/reference-architectures/dmz/secure-vnet-dmz
Network Watcher	https://docs.microsoft.com/azure/network-watcher/network-watcher-monitoring-overview

Remote access

You can manage your Azure resources by using the Azure portal (<https://portal.azure.com>) or via scripting (PowerShell, CLI, ARM, Rest API). Additionally, you can use domain policies to configure domain-joined VMs. Nevertheless, you might need to sign in to your Azure VMs for administrative purposes. There are several options available to do this in Azure, which you can see in Table 2-19.

Table 2-19: Remote Access Options

Option	Comment
RDP/SSH via public IP	A public IP in Azure assigned to your VM or a load balancer in front of your VM allows access to your VM https://docs.microsoft.com/azure/virtual-machines/windows/connect-logon https://docs.microsoft.com/azure/virtual-machines/linux/ssh-from-windows
RDP/SSH via jump box	A dedicated VM with a public IP is used to sign in from the internet. In a second hop within your virtual network, you connect to your other VM(s) Alternatively, you could configure P2S VPNs to the jump box from each machine you want to use for administration of Azure VMs.
RDP/SSH via ER/VPN	No public IP required. From within your corporate network you access your VM via private IP address.
Citrix XenApps Essentials	Via Citrix, an application running either in an Azure VM or on-premises can be published. Consumers can stream the screen output of the application. For more information go to https://channel9.msdn.com/Events/Ignite/Australia-2017/INF431
RDS	Same as jump box approach but using a terminal server to allow multiple users to sign in simultaneously. For more information go to https://blogs.technet.microsoft.com/hybridcloudbp/2017/01/09/quickly-deploy-rds-2016-in-azure/

Enterprise customers usually prefer to connect to Azure virtual networks via hybrid networking, hence enabling access through private IP addresses. This approach minimizes the attack vector because there are no additional public endpoints required.

You should protect all options based on a public IP address appropriately; for example, locate a jump box in a dedicated subnet and route all traffic between the jump box and other subnets via a firewall. In case you are using a proxy on your corporate internet break-out, make sure to configure RDP/SSH connections to Azure.

Application design

Application design is a huge topic, which we do not cover in detail in this document. The intent here is to give you some idea as to what you need to consider when designing applications for the cloud.

Most applications today are not simple. They might consist of many separate features that are implemented as services, components, third-party plug-ins, and other systems or resources. Integrating these items when all of the components are hosted locally in your datacenter is not a trivial task, and it can become even more of a challenge when you move your applications to a cloud-based environment.

Microservices versus monolithic applications

Classically, applications are broken down into three tiers: UI, business logic, and serialization. Within these three tiers, different components or functions serve different use cases. When it comes to scale, you can either scale-up within a tier or scale-out, adding additional servers to one of the tiers. If you need to add new functionality or fix a bug, the entire tier usually is updated, including various types of testing. When using microservices, the components within a classic tier are more loosely coupled, each having its own unique communication endpoint. This makes it possible for code and state to be individually versioned, deployed, and scaled, as illustrated in Figure 2-14.

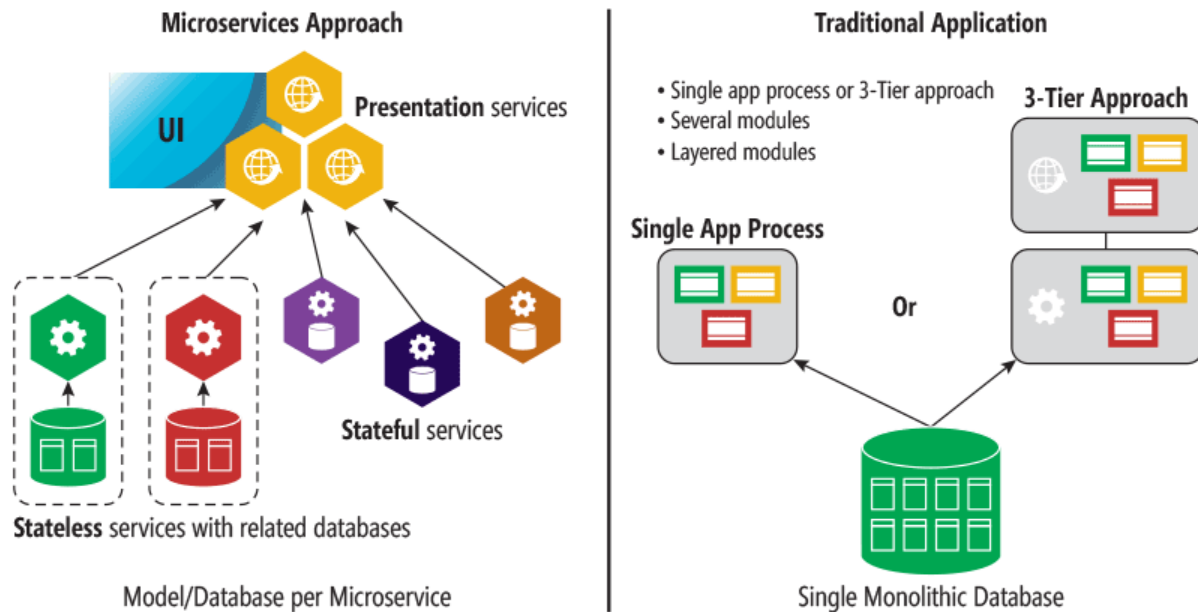


Figure 2-14: Monolithic versus microservices architecture

Dealing with unexpected failures is one of the most difficult problems to solve, especially in a distributed system. What happens when the machine on which the microservice is running fails? We schedule microservices onto different compute nodes, use naming services so that they can locate one another. We can move microservice instances if they die, the compute node dies, or for capacity reasons. Ideally, a microservice is resilient to failures in dependent services and does not itself die when a dependent service fails.

Here are some of the more common problems that an application architect still needs to consider:

- Deploy and upgrade services
- Detect failed services and restart them
- Discover services and route messages
- Manage state and monitor health
- Autoscale and rebalance microservice instances

These common problems are usually addressed by an underlying infrastructure fabric for microservices such as Kubernetes, Azure Service Fabric, Cloud Foundry, OpenShift, Docker Swarm, or DC/OS.

The benefits of a microservices-based architecture come at the cost of enhanced complexity. You need to control release management on the microservices level as well as implement and operate a

microservices fabric infrastructure in addition to your application. Also, an overhead of network traffic between microservices must be considered. As a best practice, you should use a microservices-based architecture only if the gained benefits outweigh the higher complexity. In some cases, it might be sufficient to decompose a tier by using native Azure services such as Web and API Apps, Azure Functions, Storage or Service Bus Queues, and Logic Apps.

Table 2-20 provides links to resources where you can learn more about the pros and cons of microservices-based versus monolithic applications design.

Table 2-20: Microservices versus monolithic sources

Topic	Resource
Understanding microservices	https://docs.microsoft.com/azure/service-fabric/service-fabric-overview-microservices
Service Fabric and microservices	https://msdn.microsoft.com/magazine/mt595752.aspx
Cloud Foundry	https://docs.microsoft.com/azure/virtual-machines/linux/cloudfoundry-get-started
OpenShift	https://blog.openshift.com/tag/azure/ https://blog.openshift.com/tag/azure/
DC/OS	https://docs.microsoft.com/azure/container-service/dcos-swarm/

Cloud patterns

Building a reliable and performant application in the cloud is different than building it in an on-premises environment. Whereas historically you might have purchased higher-end hardware to scale up, in a cloud environment you must scale out instead of up. Costs for cloud environments are kept low by using commodity hardware. Instead of focusing on preventing failures and optimizing Mean Time Between Failures (MTBF), in this new environment, the focus shifts to Mean Time to Recovery (MTTR) with the goal of minimizing the impact of a failure.

The Table 2-21 gives a brief overview of the cloud design challenges and patterns to address them.

Table 2-21: Challenges in cloud development

Challenge	Patterns
Availability	<p>Availability can be affected by system errors, infrastructure problems, malicious attacks, and system load. Applications typically provide users with an SLA, so applications must be designed to maximize availability.</p> <ul style="list-style-type: none"> • Health Endpoint Monitoring. Implement functional checks in an application that external tools can access through exposed endpoints at regular intervals. • Queue-Based Load Leveling. Use a queue that acts as a buffer between a task and a service that it invokes to smooth intermittent heavy loads. • Throttling. Control the consumption of resources used by an instance of an application, an individual tenant, or an entire service.

Challenge	Patterns
Data management	<p>Data management influences most of the quality attributes. Data is typically hosted in different locations and across multiple servers for reasons such as performance, scalability, or availability, and this can present a range of challenges. For example, data consistency must be maintained, and data will typically need to be synchronized across different locations.</p> <ul style="list-style-type: none"> • Cache-Aside. Load data on demand into a cache from a data store • CQRS. Segregate operations that read data from operations that update data by using separate interfaces. • Event Sourcing. Use an append-only store to record the full series of events that describe actions taken on data in a domain. • Index Table. Create indexes over the fields in data stores that are frequently referenced by queries. • Materialized View. Generate prepopulated views over the data in one or more data stores when the data isn't ideally formatted for required query operations. • Replication. Store multiple copies of each partition for durability and performance (if secondary copies are readable). • Sharding. Divide a data store into a set of horizontal partitions or shards. • Static Content Hosting. Deploy static content to a cloud-based storage service that can deliver them directly to the client. • Valet Key. Use a token or key that provides clients with restricted direct access to a specific resource or service.
Design and implementation	<p>Good design encompasses factors such as consistency and coherence in component design and deployment, maintainability to simplify administration and development, and reusability so that components and subsystems can be used in other applications and in other scenarios. Decisions made during the design and implementation phase have a huge impact on the quality and the total cost of ownership of cloud-hosted applications and services.</p> <ul style="list-style-type: none"> • CQRS. Segregate operations that read data from operations that update data by using separate interfaces. • Compute Resource Consolidation. Consolidate multiple tasks or operations into a single computational unit • External Configuration Store. Move configuration information out of the application deployment package to a centralized location. • Leader Election. Coordinate the actions performed by a collection of collaborating task instances in a distributed application by electing one instance as the leader that assumes responsibility for managing the other instances. • Pipes and Filters. Break down a task that performs complex processing into a series of separate elements that can be reused.

Challenge	Patterns
	<ul style="list-style-type: none"> • Runtime Reconfiguration. Design an application so that you can reconfigure it without requiring redeployment or restarting the application. • Static Content Hosting. Deploy static content to a cloud-based storage service that can deliver them directly to the client.
Messaging	<p>The distributed nature of cloud applications requires a messaging infrastructure that connects the components and services, ideally in a loosely coupled manner in order to maximize scalability. Asynchronous messaging is widely used, and provides many benefits, but also brings challenges such as the ordering of messages, poison message management, idempotency, and more.</p> <ul style="list-style-type: none"> • Competing Consumers. Facilitate multiple concurrent consumers to process messages received on the same messaging channel. • Pipes and Filters. Break down a task that performs complex processing into a series of separate elements that can be reused. • Priority Queue. Prioritize requests sent to services so that requests with a higher priority are received and processed more quickly than those with a lower priority. • Queue-Based Load Leveling. Use a queue that acts as a buffer between a task and a service that it invokes in order to smooth intermittent heavy loads. • Scheduler Agent Supervisor. Coordinate a set of actions across a distributed set of services and other remote resources.
Managing and monitoring	<p>Cloud applications run in a remote datacenter where you do not have full control of the infrastructure or, in some cases, the operating system. Applications must expose runtime information that administrators and operators can use to manage and monitor the system as well as support changing business requirements and customization without requiring the application to be stopped or redeployed.</p> <ul style="list-style-type: none"> • External Configuration Store. Move configuration information out of the application deployment package to a centralized location. • Health Endpoint Monitoring. Implement functional checks in an application that external tools can access through exposed endpoints at regular intervals. • Runtime Reconfiguration. Design an application so that it can be reconfigured without requiring redeployment or restarting the application.
Performance and scalability	<p>Cloud applications typically encounter variable workloads and peaks in activity. Applications should be able to scale out within limits to meet peaks in demand, and then scale in when demand decreases. Scalability concerns not just compute instances, but other elements such as data storage, messaging infrastructure, and more.</p> <ul style="list-style-type: none"> • Cache-Aside. Load data on demand into a cache from a data store • CQRS. Segregate operations that read data from operations that update data by using separate interfaces. • Event Sourcing. Use an append-only store to record the full series of events that describe actions taken on data in a domain.

Challenge	Patterns
	<ul style="list-style-type: none"> • Index Table. Create indexes over the fields in data stores that are frequently referenced by queries. • Materialized View. Generate prepopulated views over the data in one or more data stores when the data isn't ideally formatted for required query operations. • Priority Queue. Prioritize requests sent to services so that requests with a higher priority are received and processed more quickly than those with a lower priority. • Queue-Based Load Leveling. Use a queue that acts as a buffer between a task and a service that it invokes in order to smooth intermittent heavy loads. • Replication. Store multiple copies of each partition for durability and performance (if secondary copies are readable). • Sharding. Divide a data store into a set of horizontal partitions or shards. • Static Content Hosting. Deploy static content to a cloud-based storage service that can deliver them directly to the client. • Throttling. Control the consumption of resources used by an instance of an application, an individual tenant, or an entire service.
Resiliency	<p>Resiliency is the ability of a system to gracefully handle and recover from failures. The nature of cloud hosting, where applications are often multitenant, use shared platform services, compete for resources and bandwidth, communicate over the internet, and run on commodity hardware means that there is an increased likelihood that both transient and more permanent faults will arise. Detecting failures and recovering quickly and efficiently is necessary to maintain resiliency.</p> <ul style="list-style-type: none"> • Circuit Breaker. Handle faults that might take a variable amount of time to fix when connecting to a remote service or resource. • Compensating Transaction. Undo the work performed by a series of steps. • Health Endpoint Monitoring. Implement functional checks in an application that external tools can access through exposed endpoints at regular intervals. • Leader Election. Coordinate the actions performed by a collection of collaborating task instances in a distributed application by electing one instance as the leader that assumes responsibility for managing the other instances. • Queue-Based Load Leveling. Use a queue that acts as a buffer between a task and a service that it invokes to smooth intermittent heavy loads. • Retry. Make it so that an application can handle anticipated, temporary failures when it tries to connect to a service or network resource by transparently retrying an operation that's previously failed. • Scheduler Agent Supervisor. Coordinate a set of actions across a distributed set of services and other remote resources.

Challenge	Patterns
Security	Security is the capability of a system to prevent malicious or accidental actions outside of the designed usage and to prevent disclosure or loss of information. Cloud applications are often open to the public and might serve untrusted users. Applications must be designed and deployed in a way that protects them from malicious attacks, restricts access to only approved users, and protects sensitive data.

Note that resiliency is the ability to recover from failures and continue to function. It's not about avoiding failures, but responding to failures in a way that avoids downtime or data loss. The goal of resiliency is to return the application to a fully functioning state after a failure. Two important aspects of resiliency are high availability and disaster recovery.

High availability

High availability (HA) is the ability of the application to keep running in a healthy state, without significant downtime. By "healthy state," we mean the application is responsive, and users can connect to it and interact with it.

Disaster recovery

Disaster recovery (DR) is the ability to recover from rare but major incidents; for instance, nontransient, wide-scale failures, such as service disruption that affects an entire region. Disaster recovery might include data backup and archiving and can include manual interventions such as restoring a database from backup.

Resiliency is not an add-on. It must be designed into the application and put into operational practice. We recommend following this general model when designing a resilient cloud app:

- *Define* your availability requirements, based on business needs.
- *Design* the application for resiliency. Start with an architecture that follows proven practices, and then identify the possible failure points in that architecture.
- *Implement* strategies to detect and recover from failures.
- *Test* the implementation by simulating faults and triggering forced failovers.
- *Deploy* the application into production using a reliable, repeatable process.
- *Monitor* the application to detect failures. By monitoring the system, you can gauge the health of the application and respond to incidents if necessary.
- *Respond* if there are incidents that require manual interventions.

We recommend that you also use the Azure Architecture [resiliency checklist](#) for more detailed design considerations of your cloud app before you begin to code.

In Azure, the SLA describes Microsoft's commitments for uptime and connectivity. You should define your own target SLAs for each workload in your solution. Think about the time window against which your SLA is measured. The smaller the window, the tighter the tolerances. It probably doesn't make sense to define your SLA in terms of hourly or daily uptime. A good practice is to work with monthly availability.

Also browse through the [Azure reference architectures](#). We will add additional reference architectures and best practices to this list over time.

Table 2-22 lists additional resources on application design.

Table 2-22: Application design resources

Topic	Resource
Cloud design patterns	https://docs.microsoft.com/azure/architecture/patterns/
Availability patterns	https://docs.microsoft.com/azure/architecture/patterns/category/availability
Data management patterns	https://docs.microsoft.com/azure/architecture/patterns/category/data-management
Design and implementation patterns	https://docs.microsoft.com/azure/architecture/patterns/category/design-implementation
Messaging patterns	https://docs.microsoft.com/azure/architecture/patterns/category/messaging
Management and monitoring patterns	https://docs.microsoft.com/azure/architecture/patterns/category/management-monitoring
Performance and scalability patterns	https://docs.microsoft.com/azure/architecture/patterns/category/performance-scalability
Designing resilient applications for Azure	https://docs.microsoft.com/azure/architecture/resiliency/index#resiliency-strategies
Resiliency patterns	https://docs.microsoft.com/azure/architecture/patterns/category/resiliency
Resiliency checklist	https://docs.microsoft.com/azure/architecture/checklist/resiliency?toc=/azure/architecture/resiliency/toc.json
Security patterns	https://docs.microsoft.com/azure/architecture/patterns/category/security
Azure reference architectures	https://docs.microsoft.com/azure/architecture/reference-architectures/
Testing performance of a cloud service	https://docs.microsoft.com/azure/vs-azure-tools-performance-profiling-cloud-services
Load testing with Visual Studio	https://docs.microsoft.com/azure/service-fabric/service-fabric-vs-load-test
Pen testing your Azure application	https://blogs.msdn.microsoft.com/azuresecurity/2015/08/03/pen-testing-your-applications-in-microsoft-azure/
Pen testing from Azure VMs	https://blogs.msdn.microsoft.com/azuresecurity/2016/08/29/pen-testing-from-azure-virtual-machines/

Application development and operations

Before we can talk about operations, we should look at the application landscape and the organization that is common for most enterprises.

Traditionally, enterprise IT organizations have their own datacenter infrastructure running the following:

- Standard applications such as databases, web servers, middleware, or Enterprise Resource Planning (ERP) products
- Business applications, representing an important part of the organizations knowledge

These types of applications are primarily accessed by PCs and laptops. The new default in the digital world is to create and run business applications on top of a cloud platform such as Microsoft Azure; standard applications are consumed as platform as a service (PaaS) or software as a service (SaaS), and these applications are accessed by PCs, laptops, tablets, and phones. The reality for most organizations is that there is a mix of traditional on-premises applications and new, modern cloud applications, and operations must deal with that.

Companies often are organized similar to the depiction in Figure3-1.

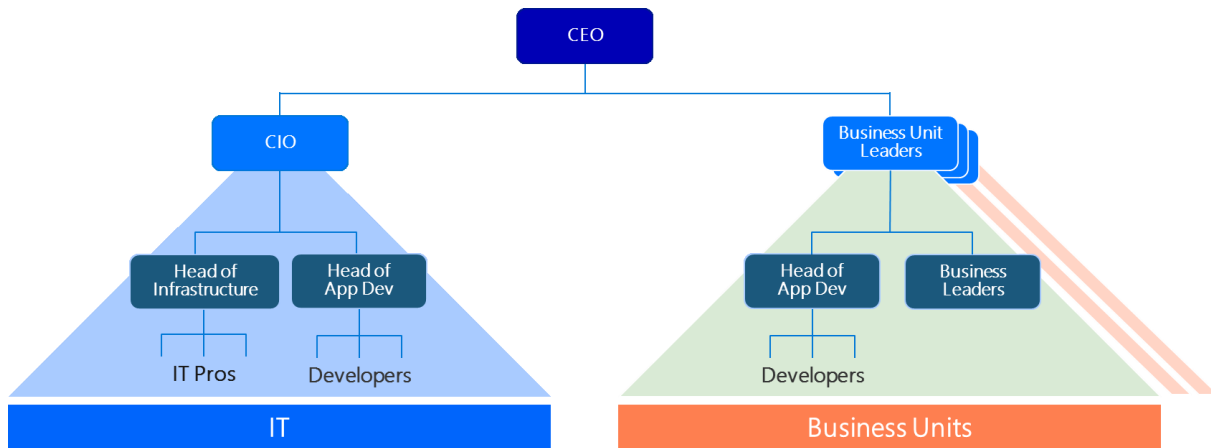


Figure 3-1: A common corporate organization

The IT organization is headed by a CIO, and there is often a split in IT between the heads of infrastructure and app development. In the business units, there are typically leaders, the business leaders, and, again, a head of application development.

The role of applications is dramatically changing as part of digital transformation (Figure 3-2). Today, every company is becoming a software company. Applications are no longer just support, they are key differentiators. They are an important part of the business. This also means that the mindset to produce quality software has come to dominate.



Figure 3-2: Growing importance of software

Independent of the approach that you take for creating and running applications, you need to think about the following three aspects:

- **Governance.** Making decisions about the application
- **Development.** Creating and updating the application
- **Operations.** Deploying and managing the application

Business application development

The ability to develop and deliver software is an important piece of any organization's ability to deliver value to customers, pivot when necessary, beat competitors to market, and respond to regulatory and compliance requirements. Delivering value with software often requires a technology transformation, and these transformations necessitate improving key capabilities.

Waterfall to Agile to DevOps

In the past, the most in-demand approach to project management was the *Waterfall* approach. This is a linear and sequential approach that had separately set goals for each defined phase of the project: requirement definition, software design, implementation, testing, deployment, maintenance, and retirement. The entire process of software development was divided into distinct processes, each having its own beginning and end, and each cascaded into the next process in a linear, “waterfall” fashion, as illustrated in Figure 3-3.

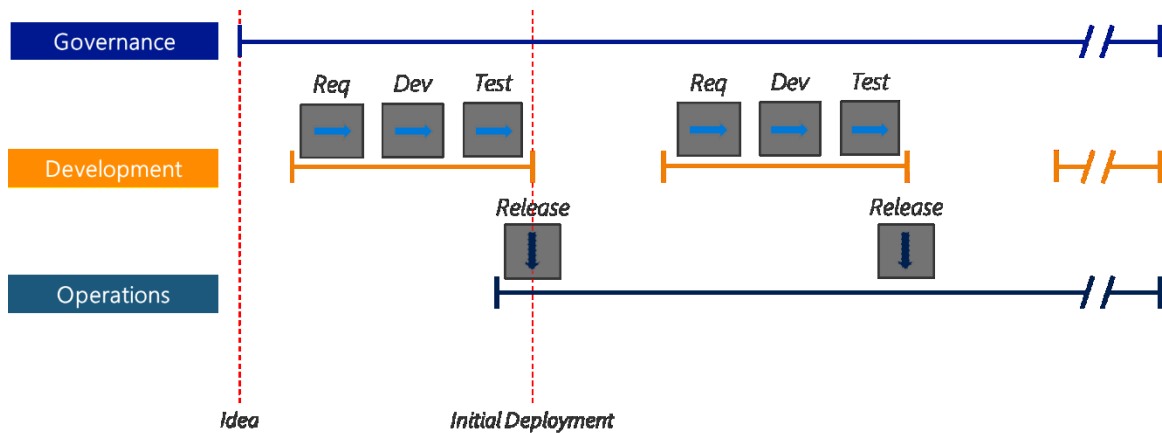


Figure 3-3: Waterfall approach to software development

But with the growing complexities and variations of the IT world, there came a need for a change in this typical approach. The *Agile* software methodology, shown in Figure 3-4, evolved as an incremental model in which the software is developed, tested, and implemented in incremental and rapid cycles. The results are incremental releases, with each release depending upon the previous release’s operations and success ratio. The primary advantage is that errors and loopholes are addressed before the project goes live. But the time between the releases didn’t typically change in comparison to the Waterfall approach.

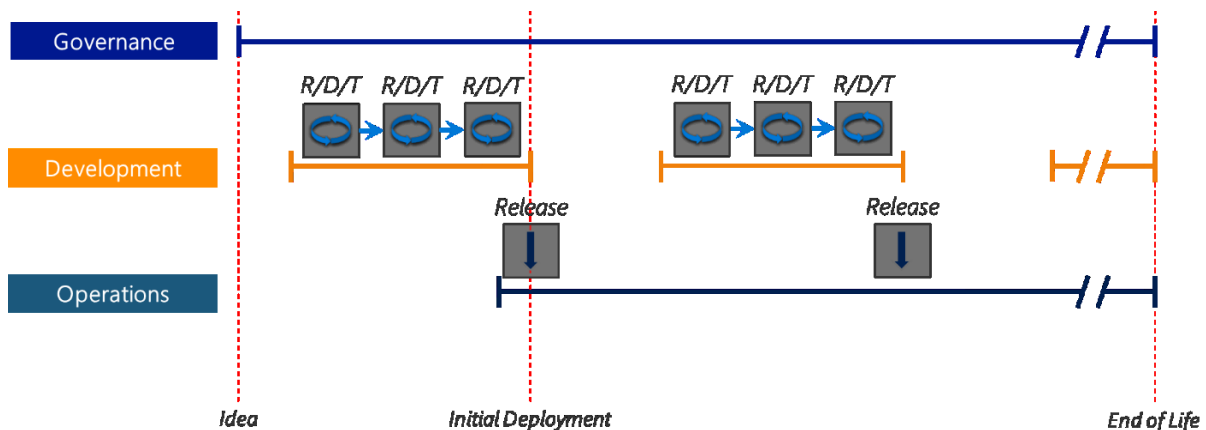


Figure 3-4: Agile approach to software development

Still, some of the key business requirements, such as faster time to market and enhanced agility weren’t, being fulfilled by either of these approaches. Furthermore, the handoff from development to operations was very difficult and time consuming, and the knowledge from the development team never really moved to the operations team. Because of that, the development team continued to be involved during operations (change management, incident management, etc.).

In both approaches, developers didn't concern themselves with how to deploy a product or how to promote it in the target environment; developers were supposed to produce the code, implement new features, and not bother with its delivery. Usually, developers wrote code on desktop computers and had little or no interest in how production servers were configured. For its part, an operations team had no influence upon a development team and didn't worry about product development. Furthermore, operations and development teams belonged to different departments with different managers and success criteria.

Organizations regularly faced situations in which the provided code did not work in the staging environment or an operations team was not able to deploy it. The development team was armored with the same rejoinder: "But it was working on my workstation!"

Operations blamed development, development blamed operations: it was a nonstop battle between the two teams leading to unpredictable delivery dates and all of the further negative consequences.

We can summarize the problems with those approaches into the following issues:

- There is no single responsible person who would manage the product from definition of business requirements to the product release.
- The development and operations teams have different success metrics. This leads to an environment in which each team is interested only in its own success.
- Lack of communication between the teams. Developers need more knowledge about the target environment, whereas operations have no clue what a development team does.
- There is a difference between development and target environment configurations.
- Slow and long delivery processes with unpredictable delivery date.

DevOps was a response to the growing awareness that there has been a disconnect between what is traditionally considered a development activity and what is traditionally considered an operations activity. The development crew is employed to respond to change in an organization, and thus it is the goal of that team to ensure that there are continuous changes. On the other hand, the operational crew has a view of change being an enemy, owing to the fact that the business depends on operations to keep the lights on and maintain stability and reliability in the organization. Apparently, there are two areas of interest in a team that will, no doubt, create a wall of confusion between these two departments; hence, the need for DevOps (Figure 3-5) to break down this wall and instead create a bridge that will allow for flow and quick deployments of deliverables in each life cycle. (See Figure 3-6.)

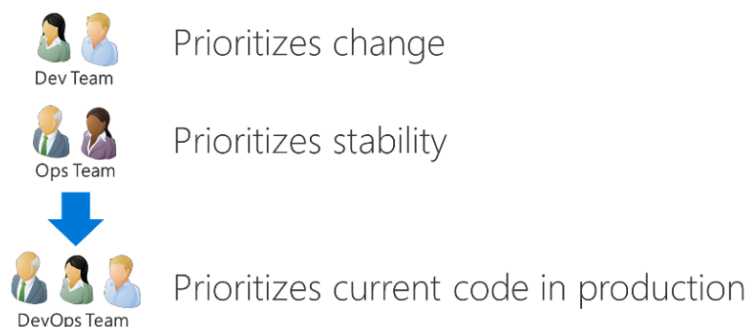


Figure 3-5: Changes versus stability

Following are the key characteristics:

- Infrastructure as code
- Continuous Delivery and Continuous Integration
- Automated deployment pipeline
- Cross-functional team and skills
- Optimized and standardized utilization of toolsets
- A culture that makes this possible

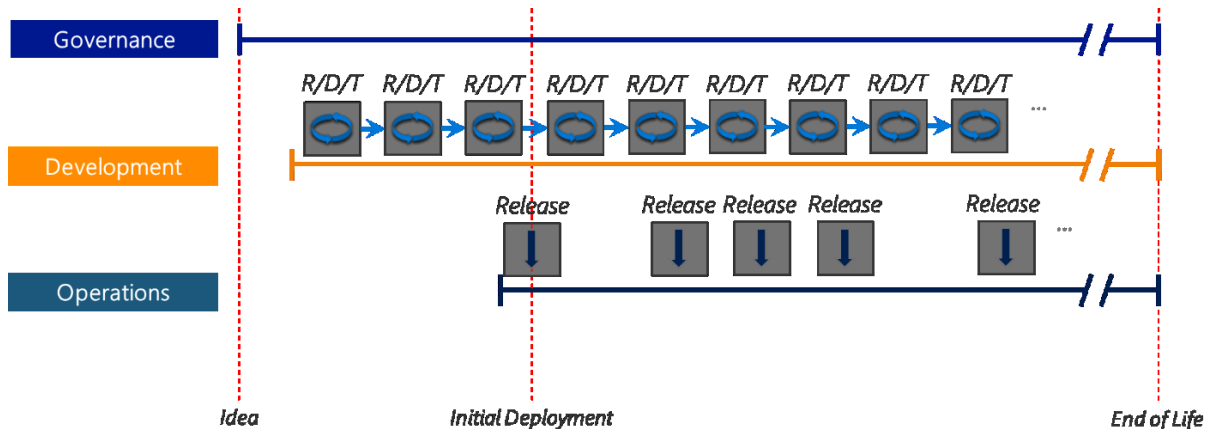


Figure 3-6: The DevOps approach

Governance is often left out of DevOps, but it also needs attention. DevOps is fundamentally changing the funding model for application development (Figure 3-7). In Waterfall and Agile development projects, we had predefined releases with a concrete set of functions and an estimated budget to deliver those. The business granted the budget separately for every release. In a DevOps model with Continuous Delivery and Continuous Integration, there must be continuous funding for the application development.

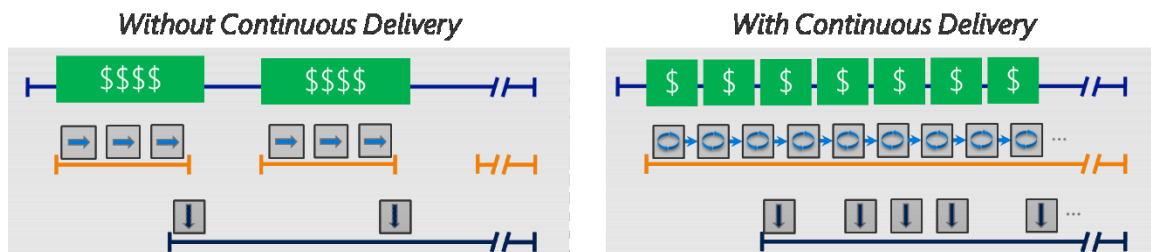


Figure 3-7: DevOps funding

DevOps is the most modern approach. It makes the business happier due to more frequent releases of an application and should be the default for all new software development projects, wherever possible.

DevOps

DevOps is the union of people, process, and products to make possible the continuous delivery of value to your end users. The contraction of “Dev” and “Ops” refers to replacing siloed development and operations to create multidisciplinary teams that now work together with shared and efficient practices and tools. Essential DevOps practices include Agile planning, Continuous Integration,

Continuous Delivery, and monitoring of applications. DevOps makes it possible for teams to deliver more secure, higher-quality solutions faster and cheaper.

Customers expect a dynamic and reliable experience when consuming software and services. Teams must rapidly iterate on software updates, measure the impact of the updates, and respond quickly with new development iterations to address issues or provide more value. Azure has removed traditional bottlenecks and helped commoditize infrastructure. Software reigns in every business as the key differentiator and factor in business outcomes. No organization, developer, or IT worker can or should avoid the DevOps movement.

Mature DevOps practitioners adopt several of the practices presented in the list that follows. These practices involve people to form strategies based on the business scenarios. Tooling can help automate the various practices.

- **Agile planning and project management.** These techniques are used to plan and isolate work into *sprints*, manage team capacity, and help teams quickly adapt to changing business needs.
- **Version control (usually with Git).** This makes it possible for teams located anywhere in the world to share source code and integrate with software development tools to automate the release pipeline.
- **Continuous Integration.** This drives the ongoing merging and testing of code, which leads to finding defects early. Other benefits include less time wasted on fighting merge issues and rapid feedback for development teams.
- **Continuous Delivery.** The continuous delivery of software solutions to production and testing environments helps organizations to quickly fix bugs and respond to ever-changing business requirements.
- **Monitoring.** Keeping a close eye on running applications, including production environments for application health as well as customer usage, helps organizations form a hypothesis and quickly validate or disprove strategies. Rich data is captured and stored in various logging formats.
- **Infrastructure as Code.** This practice facilitates the automation and validation of creation and teardown of networks and virtual machines (VMs) to help with delivering secure, stable application hosting platforms.
- **Configuration as Code.** Not only virtual machines and PaaS services are defined as code, but also the configuration of these resources. Azure Desired State Configuration, Chef, Puppet, and so on are used to define what services should be running in a VM and what configuration these services must have—in other words, a web server listening to a specific host name—as well as the baseline security configuration of the guest operating system (OS).
- **Security as Code.** In the build process, all kinds of tests are automated. Security tests are integrated to identify security issues as early as possible. For more details, see [Secure DevOps Kit for Azure](#).
- **Microservices.** This application architecture isolates business use cases into small reusable services that affords scalability and efficiency.

The following sections explain the DevOps culture and processes (based on some publications written by Sam Guckenheimer, Partner PM Manager at Microsoft) without focusing on specific technologies. You can achieve DevOps on Azure by using the Microsoft ecosystem or by using the open-source ecosystem. Table 3-1 provides links where you can find more information.

Table 3-1: DevOps resources

Topic	Resource
Azure DevOps integrations	https://azure.microsoft.com/try/devops/
Complete DevOps solution	https://www.visualstudio.com/team-services/devops/
Continuous Integration and deployment	https://www.visualstudio.com/docs/build/overview
DevOps on Azure	https://azure.microsoft.com/solutions/devops/
Infrastructure as Code	https://blogs.msdn.microsoft.com/azuredev/2017/02/11/iac-on-azure-an-introduction-of-infrastructure-as-code-iac-with-azure-resource-manager-arm-template/
Configuration as Code	https://www.slideshare.net/DougSeven/devops-practicesconfiguration-as-code

DevOps culture

The DevOps culture stresses small, multidisciplinary teams, that work autonomously and take collective accountability for how actual users experience their software. For those working in DevOps, there's no place like production. Everything they do is about making the customers' live experience better.

DevOps teams apply Agile practices and include operations in the team responsibility. Teams work in small batches, focus on improving the end-to-end delivery of customer value, and strive to eliminate waste and impediments along the way. There are no silos and no blame game, because the team is mutually accountable.

DevOps teams establish a *shift-left culture*: It is desirable to fail and learn from that failure. To minimize cost of failure, it is also desirable to fail as early as possible. To identify failures as early as possible, you should run tests any time new code is submitted. As a consequence, this leads to continuous testing. The following testing principles should apply:

- You should write tests at the lowest level
- Write once, run anywhere including production system
- Product is designed for testability
- Test code is product code, only reliable tests survive
- Testing infrastructure is a shared service
- Test ownership follows product ownership
- Every closed defect should add a test

Here is an example of what your testing should cover:

- Build and unit tests
- Integration tests
- Automated acceptance tests
- User acceptance tests
- Nonfunctional tests: performance, resilience, and security tests

DevOps teams apply a growth mindset. They make beliefs explicit, hypothesize impact to create better results, and implement the hypotheses as experiments. DevOps teams use monitoring and telemetry to gather evidence in production and observe results in real time. When evidence diminishes hypotheses, the experiences become opportunities to fail fast or gather validated learning quickly from the experiment. When evidence supports hypotheses, the team uses the opportunity to persevere, or double-down, on the actions that led to improvement.

In transitioning to DevOps, teams shift their priority from optimizing Mean Time Between Failure (MTBF) to Mean Time to Mitigate (MTTM) and Mean Time to Recovery (MTTR). Unlike in the past, when lengthy processes were designed to prevent changes that might lead to problems in the field, DevOps teams stress being able to move fast, understand the impact, and react quickly.

DevOps teams think in terms of competencies, not roles. Although they include both developmental and operational skills and awareness, they share responsibility for running the live site. This means that developers on the team accept responsibility for the health of the running services and will rotate time on-call. The principle is: if you build it, you run it.

Continuous Integration

Continuous Integration (CI) is the process of automating the build and testing of code every time a team member commits changes to version control. CI encourages developers to share their code and unit tests by merging their changes into a shared version-control repository after every small task completion. Committing code prompts an automated build system to grab the latest code from the shared repository and to build, test, and validate the full master branch.

CI emerged as a best practice because software developers often work in isolation, and then they need to integrate their changes with the rest of the team's code base. Waiting days or weeks to integrate code creates many merge conflicts, difficult-to-fix bugs, diverging code strategies, and duplicated efforts. With CI, the development team's code is merged to a shared version-control branch continuously to avoid these problems.

CI keeps the master branch clean. Teams can take advantage of modern version-control systems such as Git to create short-lived feature branches to isolate their work. A developer submits a "pull request" when the feature is complete, and on approval of the pull request, the changes are merged into the master branch. Then, the developer can delete the previous feature branch. Development teams repeat the process for additional work. The team can establish branch policies to ensure that the master branch meets the desired quality criteria. Teams use build definitions to ensure that every commit to the master branch sets the automated build and testing processes in motion. By implementing CI this way, bugs are caught earlier in the development cycle, which makes them less expensive to fix. Automated tests run for every build to maintain consistent quality.

Continuous Delivery

Continuous Delivery (CD; Figure 3-8) is the process of building, testing, configuring, and deploying from a build to a production environment. Multiple testing or staging environments create a release pipeline to automate the creation of infrastructure and deployment of a new infrastructure. Successive environments support progressively longer-running activities of integration, load, and user acceptance testing (UAT). CI starts the CD process, and the pipeline stages each successive environment upon successful completion of tests.

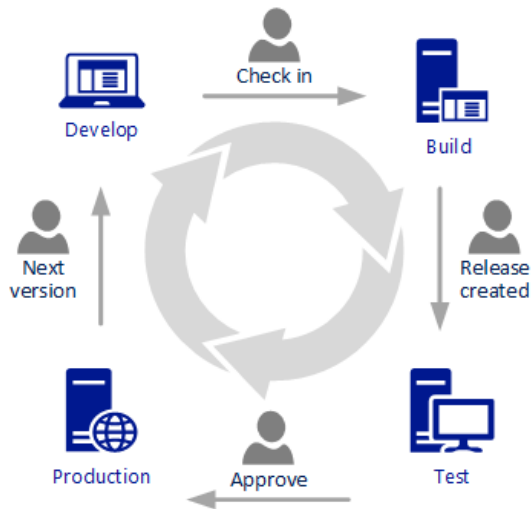


Figure 3-8: Continuous Delivery

CD might sequence multiple deployment “rings” for progressive exposure. Progressive exposure groups users who try new releases to monitor their experience in rings. The first deployment ring is often a “canary,” which you use to test new versions in production before a broader rollout. CD automates deployment from one ring to the next, which might optionally depend on an approval step, wherein a decision-maker signs off on the changes electronically. CD can create an auditable record of the approval to satisfy regulatory procedures or other control objectives.

Prior to CD, software release cycles were a bottleneck for application and operation teams. Manual processes led to unreliable releases that produced delays and errors. These teams often relied on handoffs that resulted in issues during release cycles. The automated release pipeline affords a *fail fast* approach to validation, in which the tests most likely to fail quickly are run first, and longer-running tests happen after the faster ones complete successfully.

CD is a *lean practice*. The goal of CD is to keep production fresh by achieving the shortest path from the availability of new code in version control or new components in package management to deployment. By automation, CD minimizes the time-to-deploy and time-to-mitigate (TTM) or time-to-remediate (TTR) production incidents. In lean terms, this optimizes process time and eliminates idle time.

CD is helped considerably by the complementary practices of Infrastructure as Code and Monitoring. Continuously delivering value has become a mandatory requirement for organizations. To deliver value to your end users, you must release continually and without errors.

Infrastructure as Code

Infrastructure as code (IaC) is the management of infrastructure (networks, VMs, load balancers, and connection topology) in a descriptive model, using the same versioning as the DevOps team uses for source code. Like the principle that the same source code generates the same binary, an IaC model generates the same environment every time it is applied. IaC is a key DevOps practice and is used in conjunction with CD.

IaC evolved to solve the problem of environment drift in the release pipeline. Without IaC, teams must maintain the settings of individual deployment environments. Over time, each environment becomes a “snowflake”; that is, a unique configuration that cannot be reproduced automatically. Inconsistency among environments leads to issues during deployments. With snowflakes, administration and maintenance of infrastructure involves manual processes that are difficult to track and contribute to errors.

Idempotence is a principle of IaC. Idempotence is the property that a deployment command always sets the destination environment into the same configuration, regardless of the environment's starting state. You achieve idempotency by either automatically configuring an existing destination or by discarding the existing destination and re-creating a fresh environment.

Accordingly, with IaC, teams make changes to the environment description and version the configuration model, which is typically in well-documented code formats such as JSON. The release pipeline runs the model to configure target environments. If the team needs to make changes, it edits the source, not the target.

For Azure, you can use Cloud Deployment Projects that use Azure Resource Management application programming interfaces (APIs) to create and manage Azure Resource Groups. This makes it possible for you to describe your environments with JSON. Azure Resource Groups also give you the ability to manage group-related resources together, such as websites and SQL databases. With cloud deployment projects, you can store your provisioning requirements in version control and perform Azure provisioning as part of an automated release pipeline.

With IaC, DevOps teams can test applications in production-like environments, early in the development cycle. These teams expect to provision multiple test environments reliably and on demand. You also can validate and test infrastructure represented as code to prevent common deployment issues. At the same time, the cloud dynamically provisions and tears down environments based on IaC definitions.

Configuration as Code

Configuration as Code (CaC) allows the entire configuration of your infrastructure to be stored as source code. Together with IaC, this makes it possible for the Dev team to collaborate with the Ops team on the application environments to ensure that they have the correct configuration. It also allows for continuous deployment and prevents configuration drift by idempotency.

Teams that implement IaC and CaC can deliver stable environments rapidly and at scale. They avoid manual configuration of environments and enforce consistency by representing the desired state of their environments via code. Infrastructure deployments with IaC and CaC are repeatable and prevent runtime issues caused by configuration drift or missing dependencies. DevOps teams can work together with a unified set of practices and tools to deliver applications and their supporting infrastructure rapidly, reliably, and at scale.

Security as Code

The [Secure DevOps Kit for Azure](#) is a collection of scripts, tools, extensions, automations, and so on that caters to the end-to-end Azure subscription and resource security needs for DevOps teams using extensive automation. It smoothly integrates security into native DevOps workflows, helping accomplish secure DevOps with these six focus areas:

- **Secure the subscription.** A secure cloud subscription provides a core foundation upon which you can conduct subsequent development and deployment activities. An engineering team should have the capabilities to deploy and configure security in the subscription, including elements such as alerts, Azure Resource Manager policies, role-based access control (RBAC), Security Center policies, just enough administration (JEA), and Resource Locks. Likewise, it should be possible to check that all settings are in conformance to a secure baseline.
- **Enable secure development.** During the coding and early development stages, developers should have the ability to write secure code and to test the secure configuration of their cloud applications. Just like build verification tests (BVTs), we introduce the concept of security verification tests (SVTs), which can check for security of various resource types in Azure.

- **Integrate security into CI/CD.** Test automation is a core tenet of DevOps. We emphasize this by providing the ability to run SVTs as part of the VSTS CI/CD pipeline. You can use these SVTs to ensure that the target subscription used to deploy a cloud application and the Azure resources upon which the application is built are all set up in a secure manner.
- **Continuous Assurance.** In the constantly changing DevOps environment, it is important to move away from the mindset of security being a milestone. We need to treat security as a continuously varying state of a system. This is made possible through capabilities that enable continuous assurance using a combination of automation runbooks, schedules, and so on.
- **Alerting and monitoring.** Visibility of security status is important for individual application teams and also for central enterprise teams. We provide solutions that cater to the needs of both. Moreover, the solution spans across all stages of DevOps in effect bridging the gap between the dev team and the ops team from a security standpoint through the single, integrated views it generates.
- **Cloud Risk Governance.** Lastly, underlying all activities in the kit is a telemetry framework that generates events capturing usage, adoption, evaluation results, and so forth. This allows us to make measured improvements to security targeting areas of high risk and maximum usage before others.

Monitoring

Monitoring provides feedback from production, delivering information about an application's performance and usage patterns.

One goal of monitoring is to achieve high availability by minimizing time-to-detect (TTD) and TTM. In other words, as soon as performance and other issues arise, rich diagnostic data about the issues are fed back to development teams via automated monitoring. That's TTD. DevOps teams act on the information to mitigate the issues as quickly as possible so that users are no longer affected. That's TTM. Resolution times are measured, and teams work to improve over time. After mitigation, teams work on how to remediate problems at the root cause so that those problems do not recur. That time is measured as TTR.

A second goal of monitoring is to promote *validated learning* by tracking usage. The core concept of validated learning is that every deployment is an opportunity to track experimental results that support or diminish the hypotheses that led to the deployment. Tracking usage and differences between versions gives teams the ability to measure the impact of change and drive business decisions. If a hypothesis is diminished, the team can fail fast or "pivot." If the hypothesis is supported, the team can double-down or "persevere." These data-informed decisions lead to new hypotheses and prioritization of the backlog.

Telemetry is the mechanism for collecting data from monitoring. Telemetry can use agents that are installed in the deployment environments, a software development kit (SDK) that relies on markers inserted into source code, server logging, or a combination of these. Typically, telemetry distinguishes between the data pipeline optimized for real-time alerting and dashboards and higher-volume data needed for troubleshooting or usage analytics.

Monitoring is often used to "test in production." A well-monitored deployment streams the data about its health and performance so that the team can spot production incidents immediately. Combined with a CD release pipeline, monitoring detects new anomalies and allows for prompt mitigation. This makes it possible for you to discover the "unknown unknowns" in application behavior that cannot be foreseen in preproduction environments.

Effective monitoring is essential for DevOps teams to deliver at speed, receive feedback from production, and increase customer satisfaction, acquisition, and retention.

Microservices

Microservices describes the architectural pattern of composing a distributed application from separately deployable services that perform specific business functions and communicate over web interfaces. DevOps teams encapsulate individual pieces of functionality in microservices and build larger systems by composing the microservices like building blocks, as demonstrated in Figure 3-9. Microservices apply an example of the *open/closed principle*: They are open for extension (using the interfaces they expose) and closed for modification (in that each is implemented and versioned independently). Microservices provide many benefits over monolithic architectures. They can remove single points of failure (SPOFs) by ensuring that issues in one service do not affect other parts of an application. DevOps teams can scale-out individual microservices independently to provide additional availability and capacity. Teams can extend functionality by adding new microservices without affecting other parts of the application.

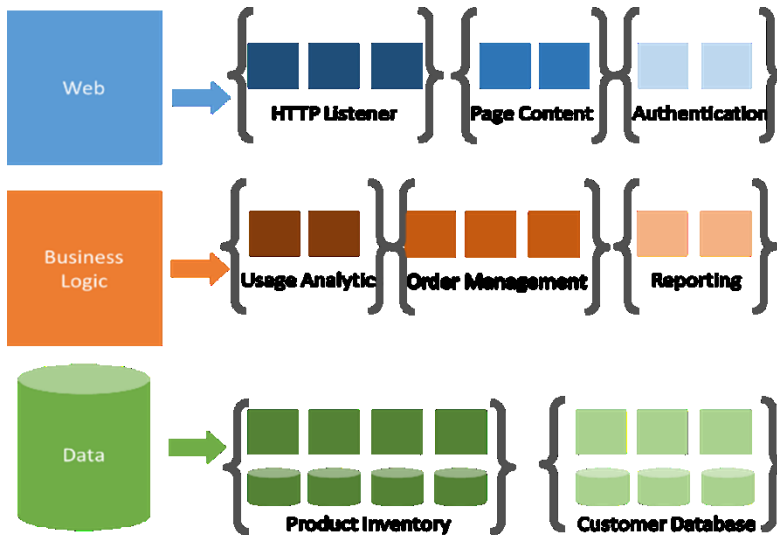


Figure 3-9: Microservices architecture

Microservices architecture can increase team velocity. DevOps practices, such as CI and CD, are used to drive microservice deployments. Microservices nicely complement cloud-based application architectures by allowing software development teams to take advantage of several patterns such as event-driven programming and autoscale scenarios. The microservice components expose APIs, typically over Representational State Transfer (REST) protocols for communicating with other services.

An emerging pattern is to use *container clusters* to implement microservices. Containers allow for the isolation, packaging, and deployment of microservices; *orchestration* scales-out a group of containers into an application.

Competencies

Nearly every IT organization wants to embrace DevOps and its promise of increased software development speed and greater business agility that results from streamlining and accelerating the interactions between development and operations.

The problem is that there's no easy or quick way to get there. A successful journey begins with the right people and the right DevOps competencies—and a willingness to collaborate. These

competencies are well explained in the online article [7 DevOps roles you need to succeed](#) and include the following:

- **DevOps evangelist.** This person must promote the benefits of DevOps by identifying and quantifying the business benefits that come from the greater agility DevOps delivers. As a change agent, the DevOps evangelist ensures buy-in from development and operational teams, identifies the key roles to support DevOps delivery methods, and makes sure IT professionals are trained and empowered to make those changes.
- **Release managers.** Release managers work to address the management and coordination of the product from development through production. Typically, they work on more of the technical details and hurdles for which a traditional project manager would not be involved. Release managers oversee the coordination, integration, and flow of development, testing, and deployment to support CD. They're focused not just on creating, but also maintaining the end-to-end application delivery tool chain.
- **Automation architects.** DevOps relies heavily on automated systems. Automation architects analyze, design, and implement strategies for CDs while ensuring high availability on production and preproduction systems. They also encompass Lean thinking across key DevOps processes.
- **Software developer-testers.** The software developer is at the heart of the DevOps organization. Under DevOps, the title of software developer might remain the same, but the new role of software developer/tester dramatically increases the scope of responsibilities. The developers are responsible not only for turning new requirements into code, but unit testing, deployment, and ongoing monitoring, as well. This shift requires a move to more automated testing so that quality doesn't suffer.
- **Experience assurance experts.** Whereas the quality assurance (QA) function is often part of software development, a new type of control becomes necessary when organizations embrace DevOps. The need for QA testers is replaced by a need for experience assurance (XA) experts who are charged with ensuring that all of new features and functions are released with the end-user's experience in mind.
- **Security engineers.** In traditional Waterfall development, system security is largely an afterthought. It's a "nonfunctional requirement" that, like quality assurance, is often tacked on at the end of system development. DevOps-minded organizations have security engineers working side by side with developers, embedding their recommendations much earlier on in the process. They build security *into* the product, not tack it on at the end.
- **Utility technology players.** Traditional IT operations professionals focus on keeping the servers running, and, in general, their machines work best when there is little change. The fast-paced DevOps environment requires a new breed. Those operations experts are now getting involved throughout the development process. It is not uncommon for these operations experts to be involved in sprint planning to ensure that improved quality of service, resource management, or security are prioritized alongside those requirements delivered from the business. DevOps requires utility team members who can operate effectively across development platforms, tools, networks, servers, and databases—even across development and support.

Checklist

DevOps is the integration of development, quality assurance, and IT operations into a unified culture and set of processes for delivering software. Use this [checklist](#) as a starting point to assess your DevOps culture and process.

Moving to DevOps

A lot of IT organizations are facing challenges in moving to a DevOps approach. The changes that these companies must make within their organizations require management buy-in, and ramping-up the new skills requires training of existing employees and probably hiring new employees. To manage the upcoming challenges in IT, you should consider making DevOps your new default:

- All new app development uses DevOps
- Make the required investment of time and money
- Adapt governance
- Change the culture: Build respect between development and operations people
- Accept the short-term losses

DevOps and classic IT service management will coexist for some time. Continue running your existing services in the classic model. But to prepare for the transition to the DevOps model, it is reasonable to apply some of the DevOps concepts to the existing development and operations of services using the Waterfall or Agile approach. Here's how to do that:

- **Deploy IaC and CaC.** One of the traditional operational challenges is automating the ability to provide appropriate environments in which to run applications and services and keeping those environments in known good states. Typically, there are environments for development, testing, acceptance, and production (DTAP). You could automate the setup of all environments by using Resource Manager templates. Check out the best practices for [Azure Resource Manager templates](#). In addition to the resources that are deployed by Resource Manager templates, it is also important to apply the same configuration automatically; that is, the same patch level of the OS of a VM, but also the same software version of an application you are using; for instance, a specific apache version and included modules along with their detailed configuration. Desired State Configuration is the Microsoft approach to automatically configure your resources and set it in a desired state. Desired State Configuration also corrects manual changes someone might apply to a machine automatically and reverts the configuration back to the desired state. There are also other products available such as Chef and Puppet that serve the same purpose and work well on Azure. To learn more, refer to the Azure Marketplace.
- **Automate your application deployments.** You should automate the deployment of new application releases for any of the aforementioned environments; for example, if your application has been approved in the test environment you should have a capability to move it automatically into the staging environment. This should also include the necessary components to monitor and back up your application. You can use technologies such as Desired State Configuration, Windows PowerShell scripts, Resource Manager, Chef, and Puppet to manage environment state and install software and dependencies into running environments.
- **Implement rich monitoring capabilities.** Begin implementing an end-to-end monitoring of your applications including health, usage, performance, security, and availability monitoring.
- **Start removing the wall between development and operations people.** Involve operations people during the development phase as early as possible to strengthen their knowledge about the application. Train them to become familiar with concepts such as IaC.

Table 3-2 provides links to resources where you can learn more about automating infrastructure and application.

Table 3-2: Automating infrastructure and application deployments

Topic	Resource
Best practices for creating Azure Resource Manager templates	https://docs.microsoft.com/azure/azure-resource-manager/resource-manager-template-best-practices
Design patterns for Resource Manager templates when deploying complex solutions	https://docs.microsoft.com/azure/azure-resource-manager/best-practices-resource-manager-design-templates
Azure Automation DSC overview	https://docs.microsoft.com/azure/automation/automation-dsc-overview
Introduction to the Azure DSC extension handler	https://docs.microsoft.com/azure/virtual-machines/windows/extensions-dsc-overview
Get started on Azure with Puppet	https://puppet.com/blog/get-started-azure-puppet
Chef Automate	https://docs.chef.io/azure_portal.html

Application operations

Distributed applications and services running in the cloud are, by their nature, complex pieces of software that comprise many moving parts. In a production environment, it's important to be able to track the way in which users employ your system, trace resource utilization, and generally monitor the health, security state, and performance of your system. You can use this information as a diagnostic aid to detect and correct issues and to help spot potential problems and prevent them from occurring.

In the cloud, you can spin up new resources within minutes. Enterprises need to keep control of these resources to keep track of costs and ensure security compliancy. This is the responsibility of a central IT custodian. The custodian needs to have an overview of all deployed resources and their security state at all times, but at the same time should not hinder modern DevOps teams from rapidly deploying new applications in an Agile way. The philosophy behind the 360° cloud management (see Figure 3-10) therefore is to gain visibility of your resources and spending as well as the health and security state of your resources. It is also possible to define policies that prevent certain configurations, such as adding a public IP to a VM.

360° of cloud management

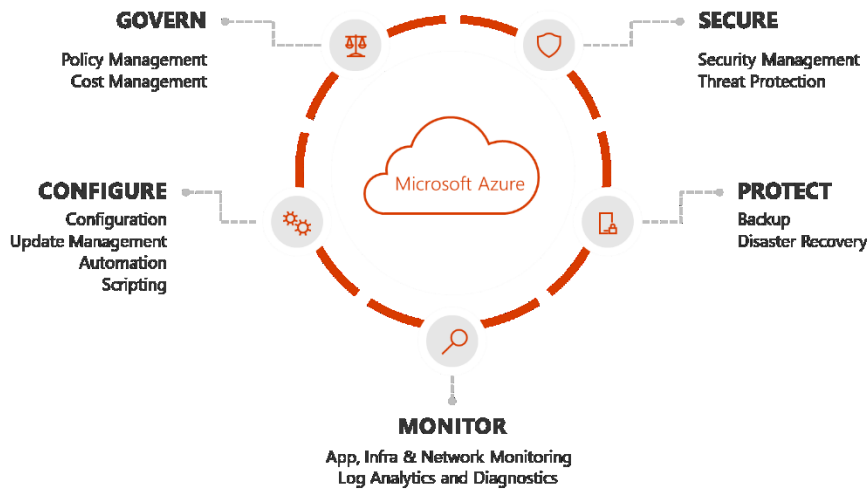


Figure 3-10: Foundation of a secure and well-managed cloud environment

Figure 3-11 illustrates a full set of built-in hybrid services to meet these needs by native, built-in Azure services and features. But you can also complement your tool landscape with other tools—for example, if you already invested in a monitoring solution or skills for a specific tool, it is absolutely fine to stick to these. Azure integrates well with the standard third-party tools. Note that Figure 3-11 shows only examples of third-party tools; the list of partner solutions is meant as an example and is far from complete.

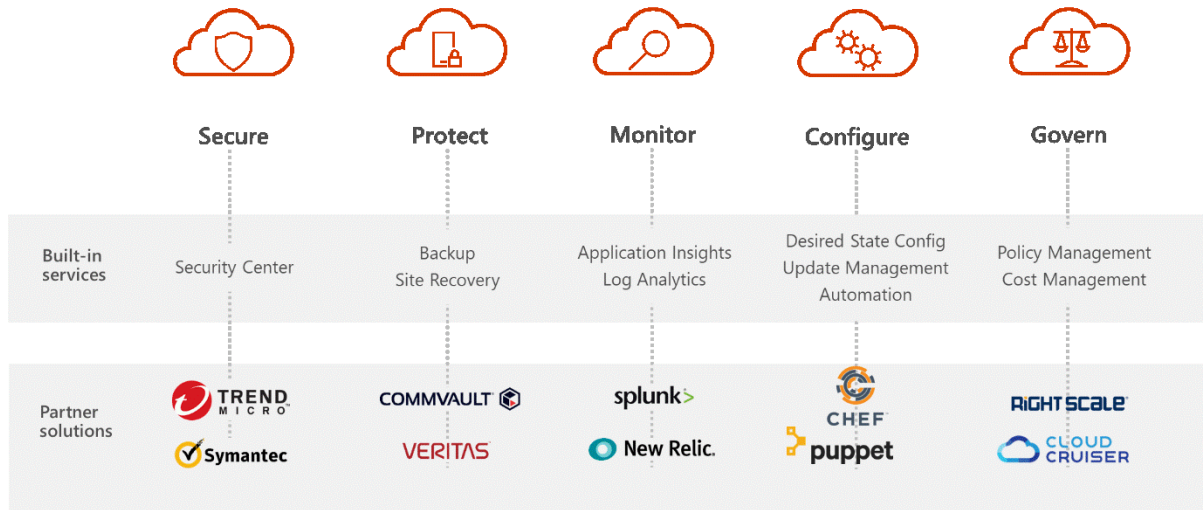


Figure 3-11: A sample of Azure built-in and complementary tools to cover 360° of cloud management

In the next subsections, we dive into the five pillars of 360° cloud management presented in Figure 3-11.

Secure

All commercial systems that include sensitive data must implement a security structure. The complexity of the security mechanism is usually a function of the sensitivity of the data. Monitoring might be able to help detect attacks on the system. For example, a large number of failed sign-in attempts might indicate a brute-force attack.

Security Center

Many organizations learn how to respond to security incidents only after suffering an attack. To reduce costs and damage, it's important to have an incident response plan in place before an attack occurs. An effective plan depends on three core capabilities: being able to protect, detect, and respond to threats. Protection is about preventing incidents, detection is about identifying threats early, and response is about evicting the attacker and restoring systems to mitigate the impacts of a breach. We strongly recommend using Security Center to gain control over the security of your Azure resources. It provides integrated security monitoring and policy management across your Azure subscriptions, helps detect threats that might otherwise go unnoticed, and works with a broad ecosystem of security solutions.

Prevent

The first step is to configure a security policy per your company's security needs. A security policy defines the set of controls that are recommended for resources within the specified subscription or resource group. We advise that you activate all areas of available recommendations of the prevention policy (e.g., System Updates, OS Vulnerabilities, Endpoint Protection, and Disk Encryption) as well as turning on data collection for each of your subscriptions to ensure that security monitoring is available for all existing and new VMs.

Security Center periodically analyzes the security state of your Azure resources. When Security Center identifies potential security vulnerabilities, it creates recommendations. The recommendations guide you through the process of configuring the needed controls. Current policy recommendations center on system updates, baseline rules, antimalware programs, network security groups on subnets and network interfaces, SQL database auditing, SQL database transparent data encryption, and web application firewalls. After reviewing all recommendations, decide which one you should apply first. We recommend that you use the severity rating as the main parameter to evaluate which recommendations you should apply before everything else.

Detect and respond

There have been significant changes in the threat landscape over the past years. In the past, companies typically had to worry only about website defacement by individual attackers who were mostly interested in seeing “what they could do.” Today’s attackers are much more sophisticated and organized. They are now interested in stealing information, financial accounts, and private data—all of which they can use to generate cash on the open market or to take advantage of a particular business, political, or military position.

Microsoft security researchers are constantly on the lookout for threats. They have access to an expansive set of telemetry gained from Microsoft’s global presence in the cloud and on-premises. This wide-reaching and diverse collection of datasets gives Microsoft a unique ability to discover new attack patterns and trends across its on-premises consumer and enterprise products as well as its online services. As a result, Security Center can rapidly update its detection algorithms as attackers release new and increasingly sophisticated exploits. This approach helps you to keep pace with a fast-moving threat environment. Security Center employs advanced security analytics, which go far beyond signature-based approaches. Breakthroughs in big data and machine learning technologies are used to evaluate events across the entire cloud fabric, detecting threats that would be impossible to identify using manual approaches and predicting the evolution of attacks.

Security Center provides you with a list of prioritized security alerts that have been detected. Each alert contains the following:

- **Description.** A brief explanation of the alert.
- **Count.** A list of all alerts of this specific type that were detected on a specific day.
- **Detected by.** The service that was responsible for raising the alert.
- **Date.** The date on which the event occurred.
- **State.** The current state for that alert. There are two types of states:
 - **Active.** The security alert has been detected.
 - **Severity.** The severity level, which can be high, medium, or low.

Additionally, Security Center can provide you with a single view of an attack campaign and all of the related alerts. This helps you to understand what actions the attacker took and what resources were affected. In Security Center, a security incident is an aggregation of all alerts for a resource that align with kill-chain patterns. Incidents appear in the Security Alerts tile and blade. An incident will reveal the list of related alerts with which you can obtain more information about each occurrence.

Protect

Most companies today have implemented a hybrid model, which means that in addition to their on-premises IT, they have a cloud footprint that has infrastructure as a service (IaaS) that possibly extends to PaaS (cloud-native applications) and SaaS (i.e., Office 365). It is important to have a consistent

experience to manage backups across the IT assets in this hybrid model. In general, there are three possible approaches backup solutions can take to use the cloud for backup solutions:

- **Cloud as storage.** In this model, the on-premises backup solution uses the public cloud as a storage destination for backup, either for the second backup copy or to replace tape backups. You still need to manage storage in the cloud, pay for any egress costs for restores, and manage the bulk of your backup infrastructure that is still on-premises.
- **Cloud as infrastructure.** This is the next level, wherein you run the backup application in an IaaS VM, which can protect applications deployed in IaaS. Although it does offer a similar experience, it can protect only IaaS VMs and not the other cloud assets (PaaS, SaaS) and is limited by the maximum amount of storage that could be attached to a VM. Also, it does not free you from infrastructure management, which is a fundamental promise of moving to the cloud.
- **Cloud as platform.** You can build backup in a PaaS model to deliver backup as a service and design it to provide a consistent management experience to both on-premises infrastructure as well as backup for cloud-native applications (IaaS, PaaS). Because all of the service infrastructure is owned and managed by the service, there would be no additional costs for the backup and there is complete freedom from managing infrastructure associated with backup.

Infrastructure and applications

Azure Backup is the Azure-based service you should use to back up and restore your data in the Microsoft cloud. Azure Backup was designed from the ground up as a PaaS service. Azure Backup offers multiple components that you download and deploy on the appropriate computer, server, or in the cloud. The component, or agent, that you deploy depends on what you want to protect. You can use the complete range of Azure Backup components (no matter whether you're protecting data on-premises or in the cloud) to back up data to an Azure Recovery Services vault.

Backup for VMs

Azure Backup seamlessly integrates with IaaS VM by providing an enable-backup experience in the VM blade itself. A VM extension is deployed when you choose to turn on Backup. You also can turn on Backup via Resource Manager templates, and it supports all the features of IaaS VMs such as drive encryption, premium drives, and so on. The foundation of Azure Backup is the Recovery Services vault. This is an online storage entity in Azure used to hold data such as backup copies, recovery points, and backup policies. Backing up VMs is a local process. You cannot back up VMs from one location to a Recovery Services vault in another location. So, for every Azure location that has VMs that you need to back up, at least one Recovery Services vault must exist in that location. With the storage replication option, you can choose between geo-redundant storage and locally redundant storage. By default, your vault has geo-redundant storage; leave this setting if this is your primary backup. Choose locally redundant storage if you want a less-expensive option that isn't quite as durable.

Before registering a VM with a vault, run the discovery process to ensure that any new VMs that have been added to the subscription are identified. The process queries Azure for the list of VMs in the subscription, along with additional information like the cloud service name and the region. Prior to your first backup, you must define a backup policy or use the default policy. The policy is the schedule for how often and when recovery points are taken. The policy also includes the retention range for the recovery points.

The Azure VM Agent must be installed on the Azure VM for the Backup extension to work. If your VM was created from the Azure gallery, the VM Agent is already present on the VM. If not—for example, you moved a VM from an on-premises datacenter—you need to install the VM Agent manually to protect the VM.

To manage the VM snapshots, the backup extension needs connectivity to the Azure public IP addresses. Without the right internet connectivity, the VM's HTTP requests time-out and the backup

operation fails. If your deployment has access restrictions in place (e.g., through a network security group), choose one of these options for providing a clear path for backup traffic:

- Add the Azure datacenter IP ranges to the safe recipients list (for instructions on how to do this, read this article)
- Deploy an HTTP proxy server for routing traffic

When the Azure Backup service initiates a backup job at the scheduled time, it signals the backup extension to take a point-in-time snapshot, as demonstrated in Figure 3-12. When the data transfer is complete, the snapshot is removed and a recovery point is created. The Azure Backup service uses the VMSnapshot extension in Windows, and the VMSnapshotLinux extension in Linux.

Backing up and restoring business-critical data is complicated by the fact that this data must be backed up while the applications that produce it are running. To address this, Azure Backup supports application-consistent backups for both Windows and Linux VMs. When taking a snapshot of Windows VMs, the Backup service coordinates with the Volume Shadow Copy Service (VSS) to get a consistent snapshot of the VM's drives. If you're backing up Linux VMs, you can write custom scripts to ensure consistency when taking a VM snapshot. Azure Backup provides a scripting framework. To ensure application consistency when backing up Linux VMs, create custom prescripts and postscripts that control the backup workflow and environment. Azure Backup invokes the prescript before taking the VM snapshot, and invokes the postscript after the VM snapshot job completes. For more details, see application consistent VM backups using prescript and postscript.

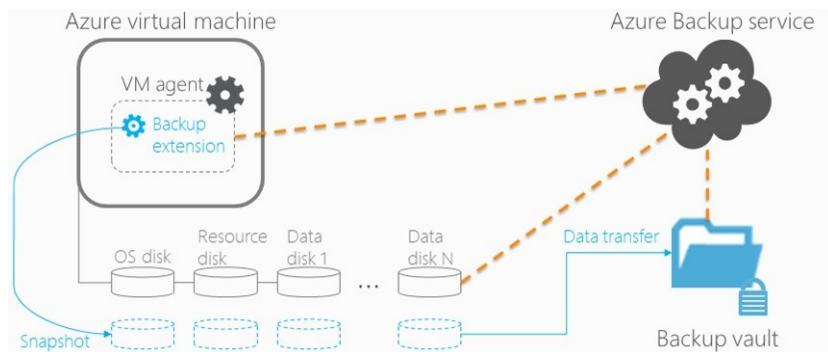


Figure 3-12: Azure Backup extension

Like backup software that is deployed on-premises, you should plan for capacity and resource utilization needs when backing up VMs in Azure. Backup data copied from a storage account adds to the input/output operations per second (IOPS) and egress (or throughput) metrics of the storage account. At the same time, VMs are also consuming IOPS and throughput. The goal is to ensure that Backup and VM traffic don't exceed your storage account limits.

Restore for VMs

Restores are based on the recovery points in the recovery services vault. If or when it is necessary to repair or rebuild a VM, you can restore the VM from any of the saved recovery points. As Figure 3-13 illustrates, when you restore a recovery point, you can create a new VM, which is a point-in-time representation of your backed-up VM, or restore drives and use the template that comes along with it to customize the restored VM or do an individual file recovery. The Azure portal provides a Quick Create option for restoring VMs. If you want to customize the VM restore configuration (e.g., special network configuration, names of created resources), you can use PowerShell.

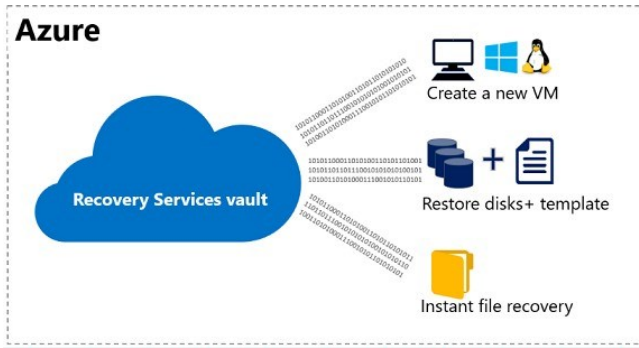


Figure 3-13: Restore virtual machines

You also can restore the drives of a backed up VM to a storage account that is in the same location as the recovery services vault. After the restoring operation is completed, you can do the following:

- Use a template to customize the restored VM
- Use the restored drives to attach to an existing VM
- Create a new VM from restored drives by using PowerShell

You can recover items such as files and folders from an Azure VM backup. This restore-as-a-service feature uses a unique approach to mount a cloud recovery point as a volume and browse it to facilitate item-level restore. You do not need to provision any infrastructure, and the egress from Azure is free. The feature is available for IaaS VMs (Windows and Linux).

Using Azure Backup, you can restore backed up VMs to the paired datacenter in the event that the primary datacenter (where your VMs are running) experiences a disaster and you configured the Backup vault to be geo-redundant. During such scenarios, you need to select a storage account, which is present in the paired datacenter and the rest of the restore process remains the same.

Azure Backup uses Compute service from the paired region to create the restored VM.

Backup and restore files and folders

With Azure Backup, you can back up your files and folders on Windows Server or Windows Clients (Figure 3-14). Therefore, you need to install the Recovery Services agent on the VM and register it with the Recovery Services vault. You can use this capability for Windows machines running in the cloud or on-premises. If required, you can provide your proxy server information to establish outbound internet connectivity from the Windows machine to the recovery service in Azure. The Microsoft Azure Backup agent provides network throttling. Throttling controls how network bandwidth is used during data transfer. This control can be helpful if you need to back up data during work hours but do not want the backup process to interfere with other internet traffic.

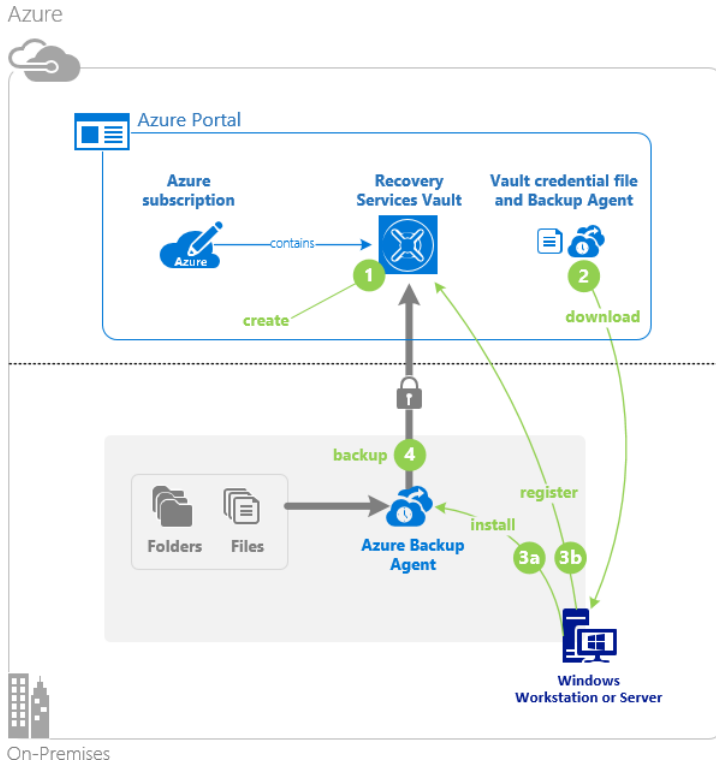


Figure 3-14: Backup and restore files and folders

Azure Backup Server and Data Protection Manager

With Azure Backup Server, you can protect application workloads such as Hyper-V VMs, VMware VMs, Microsoft SQL Server, Microsoft SharePoint Server, Microsoft Exchange, and Windows clients, all from a single console. Backup Server inherits much of the workload backup functionality from Systems Center Data Protection Manager. Although Backup Server shares much of the same functionality as Data Protection Manager, it does not back up to tape, nor does it integrate with System Center. You can install Backup Server on an Azure VM or in your datacenter. After installing and configuring the software, you need to establish the connectivity to the Recovery Services vault. Afterward, you can configure the details for the workload that you want to protect.

More info To read more, refer to [Preparing to back up workloads using Azure Backup Server](#) and [Preparing to back up workloads to Azure with DPM](#).

Table 3-3 provides additional information for protecting specific workloads.

Table 3-3: Backing up specific workloads

Topic	Resource
Back up VMware server to Azure	https://docs.microsoft.com/azure/backup/backup-azure-backup-server-vmware
Back up an Exchange server to Azure Backup with Backup Server	https://docs.microsoft.com/azure/backup/backup-azure-exchange-mabs
Back up a SharePoint farm to Azure	https://docs.microsoft.com/azure/backup/backup-azure-backup-sharepoint-mabs

Topic	Resource
Back up SQL Server to Azure using Backup Server	https://docs.microsoft.com/azure/backup/backup-azure-backup-sharepoint-mabs

Table 3-4 provides an overview of the various options to back up the aforementioned workloads.

Table 3-4: Backup overview

Component	Benefits	Limits	What is protected?	Where are backups stored?
Azure Backup (MARS) agent	<ul style="list-style-type: none"> • Back up files and folders on physical or virtual Windows OS (VMs can be on-premises or in Azure) • No separate backup server required 	<ul style="list-style-type: none"> • Backup 3x per day • Not application aware; file, folder, and volume-level restore only • No support for Linux. 	<ul style="list-style-type: none"> • Files • Folders • System State 	<ul style="list-style-type: none"> • Recovery Services vault
System Center Data Protection Manager	<ul style="list-style-type: none"> • Application-aware snapshots (VSS) • Full flexibility for when to take backups • Recovery granularity (all) • Can use Recovery Services vault • Linux support on Hyper-V and VMware VMs • Back up and restore VMware VMs using DPM 2012 R2 	<ul style="list-style-type: none"> • Cannot back up Oracle workload. 	<ul style="list-style-type: none"> • Files • Folders • Volumes • VMs • Applications • Workloads • System State 	<ul style="list-style-type: none"> • Recovery Services vault • Locally attached drive • Tape (on-premises only)

Component	Benefits	Limits	What is protected?	Where are backups stored?
Azure Backup Server	<ul style="list-style-type: none"> • App aware snapshots (VSS) • Full flexibility for when to take backups • Recovery granularity (all) • Can use Recovery Services vault • Linux support on Hyper-V and VMware VMs • Back up and restore VMware VMs • Does not require a System Center license 	<ul style="list-style-type: none"> • Cannot back up Oracle workload • Always requires live Azure subscription • No support for tape backup 	<ul style="list-style-type: none"> • Files • Folders • Volumes • VMs • Applications • Workloads • System State 	<ul style="list-style-type: none"> • Recovery Services vault • Locally attached drive
Azure IaaS VM Backup	<ul style="list-style-type: none"> • Application-aware snapshots (VSS) • Native backups for Windows/Linux • No specific agent installation required • Fabric-level backup with no backup infrastructure needed 	<ul style="list-style-type: none"> • Back up VMs once per day • Restore VMs only at disk level • Cannot back up on-premises 	<ul style="list-style-type: none"> • VMs • All drives (using PowerShell) 	<ul style="list-style-type: none"> • Recovery Services vault

More info You can find an overview on how and when to use each component in Overview of the features in Azure Backup.

Platform services

As of this writing, you cannot use Azure Backup to back up Azure Platform Services. The Azure backup service will be extended in the future for Azure SQL Database, Azure Files, and other Azure PaaS assets like Web Apps and Service Fabric for a first-class backup experience in Azure.

For the time being, you must use the native backup capabilities of the platform services, which you can see listed in Table 3-5.

Table 3-5: Backup platform services

Topic	Resource
Snapshot Blob storage	https://docs.microsoft.com/rest/api/storageservices/Snapshot-Blob
SQL Database backups	https://docs.microsoft.com/azure/sql-database/sql-database-automated-backups
Recover a database in Azure SQL Database using automated database backups	https://docs.microsoft.com/azure/sql-database/sql-database-recovery-using-backups
Automatic online backup and restore with DocumentDB	https://docs.microsoft.com/azure/documentdb/documentdb-online-backup-and-restore
Back up your app service in Azure	https://docs.microsoft.com/azure/app-service-web/web-sites-backup
Restore an app in Azure	https://docs.microsoft.com/azure/app-service-web/web-sites-restore
Protect data in Data Lake	https://docs.microsoft.com/azure/data-lake-store/data-lake-store-troubleshooting-guidance

Service configuration

IT organizations that have a DevOps-based approach for application development will deploy their infrastructure as code. The code, therefore, is managed in a source code repository, and re-creating the same infrastructure in case of any issues with the current infrastructure is straightforward.

Using Azure Resource Manager, an IT organization can repeatedly deploy an application and its infrastructure and have confidence that resources are deployed in a consistent state.

IT organizations that haven't yet adopted the DevOps model should consider exporting the configuration of their services on a regular basis. With Resource Manager, you can export a Resource Manager template from existing resources in your subscription. You can use that generated template to automate the redeployment of your solution as needed. It is important to note that there are two different ways to export a template:

- You can export the actual template that you used for a deployment. The exported template includes all the parameters and variables exactly as they appeared in the original template. This approach is helpful when you have deployed resources through the portal.
- You can export a template that represents the current state of the resource group. The exported template is not based on any template that you used for deployment. Instead, it creates a template that is a snapshot of the resource group. The exported template has many hard-coded values and probably not as many parameters as you would typically define. This approach is useful when you have modified the resource group through the portal or scripts. Now, you need to capture the resource group as a template.

More info For further details, refer to [Export an Azure Resource Manager template from existing resources](#).

Disaster recovery

Azure is divided physically and logically into units called *regions*. A region consists of one or more datacenters in close proximity. Under rare circumstances, it is possible that facilities in an entire region can become inaccessible; for example, due to network failures. Or, facilities can be lost entirely, perhaps due to a natural disaster. Fortunately, there is a lot of valuable guidance on how to design

resilient applications for Azure and how to recover from a region-wide service disruption. Protecting your stateful VMs from regional disasters and other failures was challenging in the past. Using globally redundant storage wasn't sufficient, and a disaster recovery (DR) strategy based on backup and restore couldn't always fulfill the required Recovery Point Objective (RPO) and Recovery Time Objective (RTO).

Azure Site Recovery is an Azure service that orchestrates the protection and recovery of your virtualized applications for business continuity DR (BCDR) purposes. Failover is made possible by Site Recovery, which initially copies designated VMs from a primary datacenter to the secondary datacenter and then periodically refreshes the replicas. Until now, Site Recovery supported scenarios to replicate on-premises VMs to Azure and to replicate on-premises VMs from one on-premises location to another on-premises location.

A capability has been added that supports DR of Azure VMs from one Azure region to another Azure region. Protecting your VMs is straightforward:

1. Select the source location of the VMs that you want to protect by choosing the Azure region and resource group.
2. Select the VMs within that resource group that you want to protect.
3. Define the target location and the detailed settings for the replication. By default, Site Recovery mirrors the source site configuration in the destination site by creating/using the required storage accounts, virtual network, and availability sets identified from the source. If any resource is not already available, Site Recovery will create them. But you also can customize it to change the default resource group, network, storage, and availability sets.

The first time you replicate a VM, Site Recovery creates a new replication policy with default settings of 24 hours for recovery point retention, and 4 hours for app-consistent snapshot frequency. You can adjust those settings as you need. Recovery point retention specifies the duration of the retention window for each recovery point. Protected machines can be recovered to any point within a retention window. App-consistent snapshot frequency specifies how often recovery points that contain application-consistent snapshots are created.

After the initial protection of your VMs is completed, we recommend running a test failover to validate your replication strategy or perform a DR drill without any data loss or downtime. Doing a test failover doesn't have any impact on the ongoing replication or on your production environment.

The unplanned failover option initiates the actual failover of the VM from the original Azure location to the failover location. We strongly suggest that you run a test failover before performing an unplanned failover because many of the changes under an unplanned failover are not reversible. Site Recovery will also display a warning if an unplanned failover is attempted without first running a test failover 60 days prior to the unplanned failover. In many scenarios, it is appropriate to select the option to shut down machines before beginning failover to specify that Site Recovery should try to shut down the protected VMs and synchronize the data so that the latest version of the data will be failed-over. You can use one of the following options to select your preferred recovery point:

- **Latest (default).** This option first processes all of the data that has been sent to Site Recovery service to create a recovery point for each VM before failing them over to it. This option provides the lowest RPO because the VM created after failover has all of the data that has been replicated to Site Recovery service when the failover was set off.
- **Latest processed.** This option fails-over all VMs of the recovery plan to the latest recovery point that has already been processed by Site Recovery service. If you are doing failover of a recovery plan, you can go to individual VM and look at Latest Recovery Points tile to get this information. Because no time is spent to process the unprocessed data, this option provides a low RTO failover option.

- **Custom.** If you are doing test failover of a VM, you can use this option to failover to a particular recovery point.

After you are satisfied with the failed-over VM, you can commit the failover. This deletes all the recovery points available with the service, and the Change Recovery Point option will no longer be available.

After failover is complete, the VMs start and are running at the secondary location. However, they aren't protected or replicating. When the primary site is available again with the same underlying infrastructure, you must reverse-replicate the VM. This ensures that all of the data is replicated back to the primary site, and that the VM is ready for failover again. After reprotection has been completed your VMs are ready to be failed back to the original site. This involves performing an unplanned failover in the opposite direction to that done before. After failback (failover from destination site to source site) is complete, the VMs start and are running at the secondary location. However, they aren't protected or replicating. You need to turn on replication to the destination site, following the same steps explained earlier.

Figure 3-15 depicts a typical multitier application that has a resource group, a virtual network with some subnets, a public IP address to access the applications, and each of the tiers are in an availability set with a load balancer in front of it. Multiple storage accounts are used for the various applications tiers.

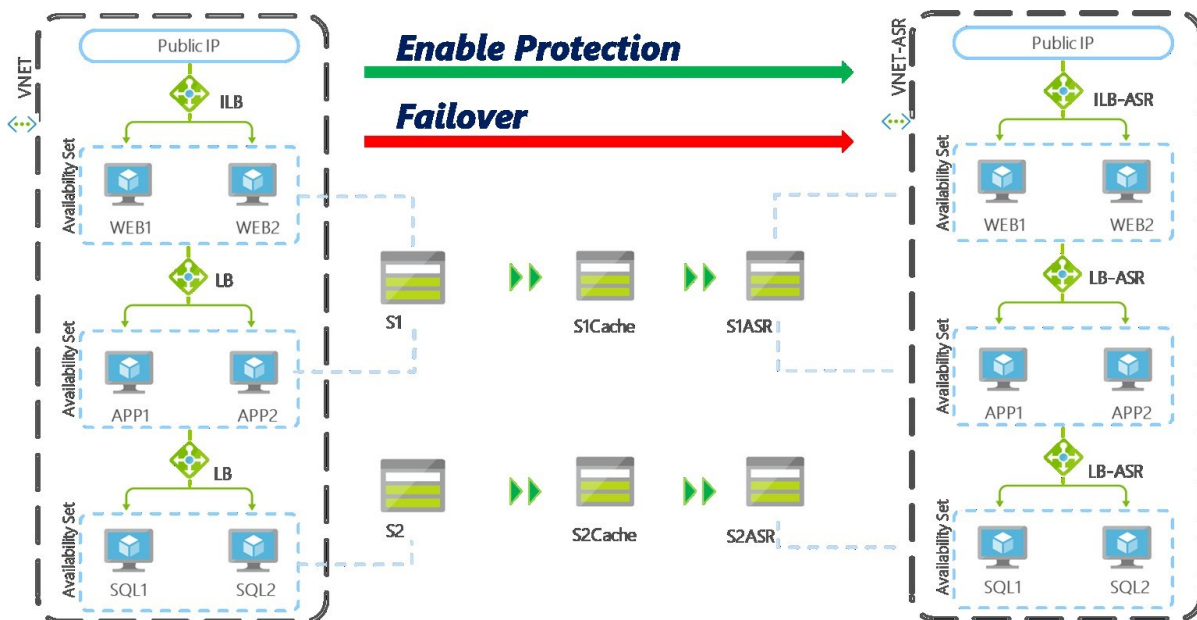


Figure 3-15: DR of Azure VMs from one Azure region to another Azure region

When you turn on replication with Site Recovery, identical resources are created on the secondary site. This includes the resource group, the virtual network, the storage accounts, and the availability sets. The data is moved from the primary storage account to a cache storage account in the same region, and from there to the secondary storage account in the other region. At the point in time when the failover happens, the VMs are created within that precreated infrastructure. Site Recovery supports the failover of multiple VMs with multiple drives in a consistent way. Site Recovery doesn't replicate network security groups, Public IP addresses, and load balancers. You must precreate those resources in the secondary location.

Before setting up Site Recovery, you need to review the following:

- Ensure that you create your Recovery Services vault in the same subscription where the Azure VMs are running. You cannot replicate VMs running in one subscription to another subscription
- We recommend that you create the Recovery Services vault in the location to which you want to replicate your machines; that is, the destination Azure location. You cannot have your vault in source location, because in the event of a region-wide disruption, your vault will also not be available
- If you are using Network Security Groups (NSG) rules to control outbound internet connectivity on the Azure VMs, ensure that you set up safe-recipient lists for the following Azure datacenter IP ranges made available in this link:
 - Source region IP ranges where your Azure VMs are running
 - Destination region IP ranges to which your VMs need to be replicated
- If you are using any firewall proxy to control outbound internet connectivity, ensure that you include them on a safe-recipient list for all the required Azure Site recovery service URLs shown in the following list or the IP ranges mentioned in the previous point.
 - ***.blob.core.windows.net.** Required so that data can be written to the storage account from the VM
 - **hypervrecoverymanager.windowsazure.com.** Required so that the Site Recovery service communication can happen from the VM
 - **169.254.169.254** Used for metadata service fetching on Azure VMs
 - **login.microsoftonline.com.** Used for authorization and authentication to the Site Recovery service URLs
- If you are setting up an Azure ExpressRoute connection between your on-premises datacenter and the Azure region and have a need for your application to communicate to the on-premises machines, ensure that you have at least "Site to Site" connection between your destination Azure region and on-premises datacenter. If a lot of traffic is expected to flow between your destination DR Azure region and on-premises datacenter, you should have another ExpressRoute connection between target Azure region and on-premises datacenter.
- If you are using Forced Tunneling between Azure Virtual Network and your on-premises datacenter, ensure that the replication traffic is not forced to on-premises by creating the correct routing rules in your forced-tunnel configuration.
- For Active Directory and DNS, it is recommended to use native Active Directory replication. You can refer to "Site to Azure" section in Protect Active Directory and DNS with Azure Site Recovery for guidance. For "Azure to Azure," best practices are like "Site to Azure." A guidance document for Active Directory and DNS for "Azure to Azure" will be published soon.
- If you are using SQL Server Always On clustering on the primary site, it is recommended to use SQL Server Always On for DR, too. You can refer to Protect SQL Server with SQL Server disaster recovery and Azure Site Recovery for guidance. Additional guidance for protecting SQL server for "Azure to Azure" will be published soon.
- Please check the Site Recovery Documentation for further guidance.

Monitor

The primary intent of IT organizations is to have end-to-end monitoring for complex business applications (as shown in Figure 3-16) that consist of Azure infrastructure and platform services as well as on-premises components.

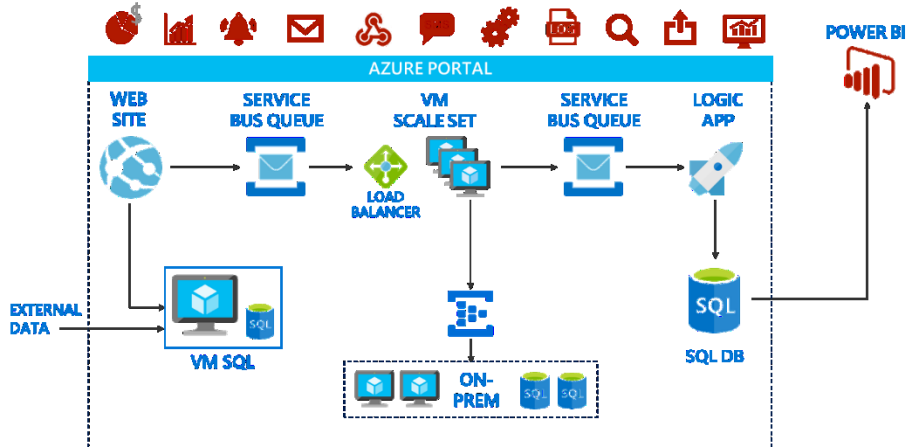


Figure 3-16: Monitoring complex applications

You can use monitoring to gain an insight into how well your application is functioning. Monitoring is a crucial part of maintaining quality-of-service targets. Common scenarios for collecting monitoring data include the following:

- **Health monitoring.** A system is healthy if it is running and capable of processing requests. The purpose of health monitoring is to generate a snapshot of the current health of the system so that you can verify that all components of the system are functioning as expected. An operator should be alerted quickly if any part of the system is deemed to be unhealthy. The operator should be able to ascertain which parts of the system are functioning normally, and which parts are experiencing problems.
- **Availability monitoring.** A truly healthy system requires that the components and subsystems that comprise the system are available. Availability monitoring is closely related to health monitoring. But whereas health monitoring provides an immediate view of the current health of the system, availability monitoring is concerned with tracking the availability of the system and its components to generate statistics about the uptime of the system. An operator should be able to view the historical availability of each system and subsystem and use this information to spot any trends that might cause one or more subsystems to periodically fail.
- **Performance monitoring.** As the system is placed under more and more stress (by increasing the volume of users), the size of the datasets that these users access grows, and the possibility of failure of one or more components becomes more likely. Frequently, component failure is preceded by a decrease in performance. If you're able to detect such a decrease, you can take proactive steps to remedy the situation.
- **Service-Level Agreement (SLA) monitoring.** Many commercial systems that support paying customers make guarantees about the performance of the system in the form of SLAs. Essentially, SLAs state that the system can handle a defined volume of work within an agreed time frame and without losing critical information. SLA monitoring is concerned with ensuring that the system can meet measurable SLAs, such as these:
 - The percentage of service uptime
 - The application throughput

- The number of successful/failed application requests
- The number of application and system faults, exceptions, and warnings
- **Usage monitoring.** Usage monitoring tracks how the features and components of an application are used. An operator can use the gathered data for the following:
 - Determine which features are heavily used and determine any potential hotspots in the system.
 - Obtain information about the operational events of the system under normal use.
 - Detect (possibly indirectly) user satisfaction with the performance or functionality of the system.
 - Generate billing information. A commercial application or multitenant service might charge customers for the resources that they use.
 - Enforce quotas. If a user in a multitenant system exceeds its paid quota of processing time or resource usage during a specified period, its access can be limited or processing can be throttled.
- **Auditing.** Depending on the nature of the application, there might be statutory or other legal regulations that specify requirements for auditing a users' operations and recording all data access.

More info To read detailed instructions, go to [Monitoring and Diagnostics](#).

Monitoring is the act of collecting and analyzing data to determine the performance, health, and availability of your business application and the resources that it depends on. An effective monitoring strategy helps you understand the detailed operation of the components of your application. It also helps you increase your uptime by proactively notifying you of critical issues so that you can resolve them before they become problems.

Azure includes multiple services that individually perform a specific role or task in the monitoring space. Together, these services deliver a comprehensive solution for collecting, analyzing, and acting on telemetry from your application and the Azure resources that support them. They can also work to monitor critical on-premises resources in order to provide a hybrid monitoring environment. Understanding the tools and data that are available is the first step in developing a complete monitoring strategy for your application.

The following diagram shows a conceptual view of the components that work together to provide monitoring of Azure resources. The following sections describe these components and provide links to detailed technical information.

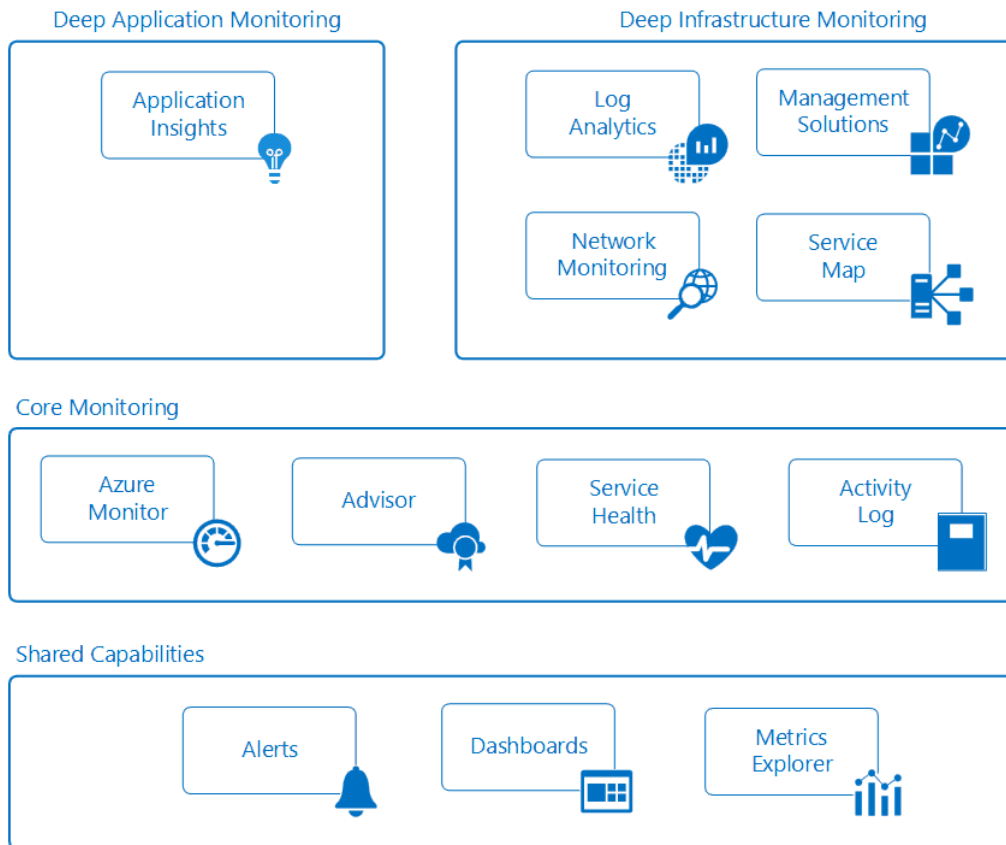


Figure 3-17: Monitoring overview

Shared Capabilities

The core and deep monitoring service share functionality which provides the following capabilities:

- **Alerts.** [Azure alerts](#) proactively notify you of critical conditions and potentially take corrective action. Alert rules can use data from multiple sources, including metrics and logs. They use [action groups](#), which contain unique sets of recipients and actions in response to an alert. Based on your requirements, you can have alerts start external actions by using webhooks and integrate with your ITSM tools.
- **Dashboards.** You can use [Azure dashboards](#) to combine different kinds of data into a single pane in the Azure portal. You can then share the dashboard with other Azure users. For example, you can create a dashboard that combines the following:
 - Tiles that show a graph of metrics
 - A table of activity logs
 - A usage chart from Application Insights
 - The output of a log search in Log Analytics

You can also export Log Analytics data to [Power BI](#). There, you can take advantage of additional visualizations. You can also make the data available to others within and outside your organization.

- **Metrics Explorer.** Metrics are numerical values generated by an Azure resource to help you understand the operation and performance of the resource. By using Metrics Explorer, you can send metrics to Log Analytics for analysis with data from other sources.

Core Monitoring

Core monitoring provides fundamental, required monitoring across Azure resources. These services require minimal configuration and collect core telemetry that the premium monitoring services use.

- **Azure Monitor** [Azure Monitor](#) enables core monitoring for Azure services by allowing the collection of [metrics](#), [activity logs](#), and [diagnostic logs](#). For example, the activity log tells you when new resources are created or modified.

Metrics are available that provide performance statistics for different resources and even the operating system inside a virtual machine. You can view this data with one of the explorers in the Azure portal and create alerts based on these metrics. Azure Monitor provides the fastest metrics pipeline (5 minute down to 1 minute), so you should use it for time-critical alerts and notifications.

You also can send these metrics and logs to Azure Log Analytics for trending and detailed analysis or create additional alert rules to proactively notify you of critical issues as a result of that analysis.

- **Azure Advisor.** [Azure Advisor](#) constantly monitors your resource configuration and usage telemetry. It then gives you personalized recommendations based on best practices. Following these recommendations helps you improve the performance, security, and availability of the resources that support your applications.
- **Service Health.** The health of your application relies on the Azure services that it depends on. [Azure Service Health](#) identifies any issues with Azure services that might affect your application. Service Health also helps you plan for scheduled maintenance.
- **Activity Log.** [Activity Log](#) provides data about the operation of an Azure resource. This information includes the following:
 - Configuration changes to the resource
 - Service health incidents
 - Recommendations on better utilizing the resource
 - Information related to autoscale operations

You can view logs for a particular resource on its page in the Azure portal. Or you can view logs from multiple resources in Activity Log Explorer.

You can also send activity log entries to Log Analytics. There, you can analyze the logs by using data collected by management solutions, agents on VMs, and other sources.

Deep monitoring services

The following Azure services provide rich capabilities for collecting and analyzing monitoring data at a deeper level. These services build on core monitoring and take advantage of common functionality in Azure. They provide powerful analytics with collected data to give you unique insights into your applications and infrastructure. They present data in the context of scenarios that are targeted to different audiences.

Deep application monitoring

It's essential to monitor a modern application while it is running. Most important, you want to detect failures before your customers do. You also want to discover and fix performance issues that, although not catastrophic, perhaps slow things down or cause some inconvenience to your users. And when the system is performing to your satisfaction, you want to know what the users are doing with it.

- **Application Insights.** You can use [Azure Application Insights](#) to monitor availability, performance, and usage of your application, whether it's hosted in the cloud or on-premises.

By instrumenting your application to work with Application Insights, you can achieve deep insights and implement DevOps scenarios. You can quickly identify and diagnose errors without waiting for a user to report them. With the information that you collect, you can make informed choices on your application's maintenance and improvements.

Application Insights has extensive tools for interacting with the data that it collects. Application Insights stores its data in a common repository. It can take advantage of shared functionality such as alerts, dashboards, and deep analysis with the Log Analytics query language.

Consider activating Application Insights (Figure 3-18) on various apps and platforms, including [Azure App](#) or [Cloud Services](#), [.NET](#), [Node.js](#), [Java](#), [JavaScript](#), and [Docker](#), hosted on-premises or in the cloud. Application Insights integrates with your DevOps process and has connection points to a variety of development tools. You install a small instrumentation package in your application and set up an Application Insights resource in Azure. The instrumentation monitors your app and sends telemetry data to the portal. You can instrument not only the web service application, but also any background components as well as the JavaScript in the web pages themselves. After the initial setup, you have a lot of options to analyze the collected data, as shown in Table 3-6.

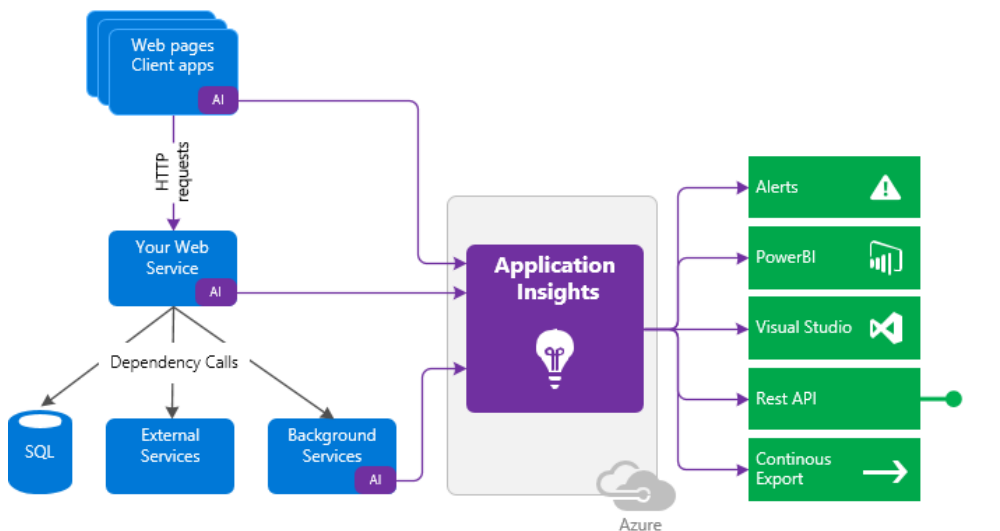


Figure 3-18: Application Insights

Table 3-6: Application Insights

Topic	Resource
Dashboards in the Azure portal	https://docs.microsoft.com/azure/application-insights/app-insights-dashboards#dashboards
Diagnostic search for instance data	https://docs.microsoft.com/azure/application-insights/app-insights-diagnostic-search
Metrics Explorer for aggregated data	https://docs.microsoft.com/azure/application-insights/app-insights-metrics-explorer

Application map	https://docs.microsoft.com/azure/application-insights/app-insights-app-map
Live Metrics Stream	https://docs.microsoft.com/azure/application-insights/app-insights-live-stream
Analytics	https://docs.microsoft.com/azure/application-insights/app-insights-analytics
Visual Studio	https://docs.microsoft.com/azure/application-insights/app-insights-visual-studio
Power BI	https://docs.microsoft.com/azure/application-insights/app-insights-export-power-bi
REST API	https://dev.applicationinsights.io/
Continuous export	https://docs.microsoft.com/azure/application-insights/app-insights-export-telemetry

Integrating your Application Insights apps to a central Log Analytics workspace increases your organization's visibility over your applications by having operation and application data in one place. [The Application Insights Connector management solution \(Preview\)](#) helps you diagnose performance issues and understand what users do with your app when it is monitored with [Application Insights](#). Views of the same application telemetry that developers see in Application Insights are available in Log Analytics. However, when you integrate your Application Insights apps with Log Analytics, visibility of your applications is increased by having operation and application data in one place. Having the same views helps you to collaborate with your app developers. The common views can help reduce the time to detect and resolve both application and platform issues.

When you use the solution, you can do the following:

- View all your Application Insights apps in a one place, even when they are in different Azure subscriptions
- Correlate infrastructure data with application data
- Visualize application data with perspectives in log search
- Pivot from Log Analytics data to your Application Insights app in the Azure portal

Deep infrastructure monitoring

- **Log Analytics.** [Log Analytics](#) plays a central role in Azure monitoring by collecting data from a variety of resources (including non-Microsoft tools) into a single repository. There, you can analyze the data by using a powerful query language.

Application Insights and Azure Security Center store their data in the Log Analytics data store and use its analytics engine. Data is also collected from Azure Monitor, management solutions, and agents installed on VMs in the cloud or on-premises. This shared functionality helps you form a complete picture of your environment.

Log Analytics requires minimal configuration and is already integrated with other Azure services. You just need to create a workspace to enable collection. You can then install agents on VMs to include them in the workspace and enable management solutions which include logic to provide additional insights into different applications. Behind the scenes, data types are either predefined or automatically created as data is collected.

Log Analytics isn't limited to monitoring Azure resources though. It can collect data from resources that are on-premises or in other clouds to create a hybrid monitoring environment and

can directly connect to System Center Operations Manager to collect telemetry from existing agents. Analysis tools in Log Analytics such as log searches, views, and management solutions work against all collected data providing you with the capability to centrally analyze your entire environment.

Log Analytics collects data from a variety of sources. After it is collected, the data is organized into separate tables for each data type, which allows all data to be analyzed together regardless of its original source.

Methods for collecting data into Log Analytics include the following:

- Configure Azure Monitor to copy metrics and logs that it collects from Azure resources.
- Agents on [Windows](#) and [Linux](#) VMs send telemetry from the guest operating system and applications to Log Analytics according to [Data Sources](#) that you configure.
- Connect a [System Center Operations Manager management group](#) to Log Analytics to collect data from its agents.
- Azure services such as [Application Insights](#) and [Security Center](#) store their data directly in Log Analytics without any configuration.
- Write data from PowerShell command line or [Azure Automation runbook](#) using Log Analytics cmdlets.
- If you have custom requirements, then you can use the [HTTP Data Collector API](#) to write data to Log Analytics from any REST API client.
- **Management solutions.** [Management solutions](#) are packaged sets of logic that provide insights for a particular application or service. They rely on Log Analytics to store and analyze the monitoring data that they collect.

Management solutions are available from Microsoft and partners to provide monitoring for various Azure and third-party services. Examples of monitoring solutions include:

- [Container Monitoring](#), which helps you view and manage your container hosts.
- [Azure SQL Analytics](#), which collects and visualizes performance metrics for Azure SQL databases.

You can view all available management solutions in the Azure Portal under the Monitor screen.

- **Network Monitoring.** There are several tools that work together to monitor various aspects of your network, whether in Azure or on-premises.
 - [Network Watcher](#) provides scenario-based monitoring and diagnostics for different network scenarios in Azure. It stores data in Azure metrics and diagnostics for further analysis. It works with the following solutions for monitoring various aspects of your network.
 - [Network Performance Monitor \(NPM\)](#) is a cloud-based network monitoring solution that monitors connectivity across public clouds, datacenters, and on-premises environments.
 - [ExpressRoute Monitor](#) is an NPM capability that monitors the end-to-end connectivity and performance over Azure ExpressRoute circuits.
 - [DNS Analytics](#) is a solution that provides security, performance, and operations-related insights, based on your DNS servers.
 - [Service Endpoint Monitor](#) tests the reachability of applications and detects performance bottlenecks across on-premises, carrier networks, and cloud/private data centers.

- **Service Map.** [Service Map](#) provides insight into your IaaS environment by analyzing virtual machines with their different processes and dependencies on other computers and external processes. It integrates events, performance data, and management solutions in Log Analytics. You can then view this data in the context of each computer and its relation to the rest of your environment.

Service Map is similar to [Application Map in Application Insights](#). It focuses on the infrastructure components that support your applications.

Examples

The following are high-level examples that illustrate how you can use different monitoring tools in Azure for different scenarios.

- **Monitoring a web application.** Consider a web application deployed in Azure through Azure App Service, Azure Storage, and a SQL database. You begin by accessing [metrics](#) and [activity logs](#) for these resources on their pages in the Azure portal. You look for critical information such as the number of requests to the application and average response time. You also identify any configuration changes.

You then go to Monitor in the portal to view metrics and logs for the different resources together. As you determine standard parameters for the metrics, you [create alert rules](#). These rules proactively notify you when, for example, average the response time increases beyond a threshold. To get a quick view of your application's daily performance, you create an Azure dashboard to show graphs of metrics that represent critical key performance indicators (KPIs).

To perform deeper monitoring of your application, you [configure it for Application Insights](#). You now can collect additional data that provides further insight into the operation and performance of your application. Application Insights detects the underlying relationships between your app's components. It allows for visual representation via [Application Map](#) coupled with [end-to-end tracing](#) to diagnose the exact component, dependency, or exception where a problem has occurred.

You create [availability tests](#) to proactively test your application from different regions. To help your developers, you [turn on the Profiler](#) so that you can track requests and any exceptions down to a specific line of code. To gain further visibility into services used in your application, you add the [SQL Analytics solution](#) to collect additional data in Log Analytics.

After some time, you decide to investigate the root cause for periods when performance on the site has fallen below a threshold. You write a query by using Log Analytics. This will help you to correlate the usage and performance data collected by Application Insights with configuration and performance data across the Azure resources that support your application.

- **Monitoring VMs.** You have a mix of Windows and Linux VMs running in Azure. You use Azure Monitor to view [activity logs](#) and [host-level metrics](#). You add the [Azure Diagnostics extension](#) to the VMs in order to collect metrics from the guest OS. You then create [alert rules](#) to proactively notify you when basic metrics like processor utilization and memory cross thresholds.

To collect more details about VMs running a business application, you [create a Log Analytics workspace and enable the VM extension](#) on each machine. You configure the [collection of different data sources](#) for your application and [create views](#) to report on its daily operation and performance. You then [create alert rules](#) to notify you when particular error events are received.

To continuously monitor the health of the installed agent, you add the [Agent Health management solution](#). To gain further insight into the application, you [add the dependency agent](#) to the VMs in order to add them to [Service Map](#). Service Map discovers critical processes and identifies connections between machines with other services.

After a reported outage, you use Service Map to perform forensics to identify the particular machines that experienced the problem. You then create a [query on the Log Analytics data](#) to identify the issue in the future. And you create an alert rule to proactively notify you when the condition is detected.

Integrate with IT Service Management

Most IT organizations already have in place an IT Service Management (ITSM) solution such as ServiceNow, Systems Center, Provance, or Cherwell. For those companies, we recommend integrating it with Log Analytics to centrally monitor and manage work items. The IT Service Management Connector provides for bidirectional integration with ITSM products, where it provides the Log Analytics users an option to create incidents, alerts, or events in an ITSM solution. The connector also imports data such as incidents and change requests from an ITSM solution into Log Analytics.

Integrate with Security Information and Event Management systems

[Azure Log Integration](#) was made available to simplify the task of integrating Azure logs with your on-premises Security Information and Event Management (SIEM) system.

The recommended method for integrating Azure logs is to use your SIEM vendor's connectors. Azure Monitor provides the ability to stream the logs into event hubs, and SIEM vendors can write connectors to further integrate logs from the event hub into the SIEM. For a description of how this works, follow the instructions [in Monitor stream monitoring for data event hubs](#). The article also lists the SIEMs for which direct Azure connectors are already available.

Azure Log Integration collects Windows events from Windows Event Viewer logs, [Azure activity logs](#), [Azure Security Center alerts](#), and [Azure Diagnostics logs](#) from Azure resources. Integration helps your SIEM solution provide a unified dashboard for all your assets, whether on-premises or in the cloud. You can use a dashboard to receive, aggregate, correlate, and analyze alerts for security events, as demonstrated in Figure 3-19.

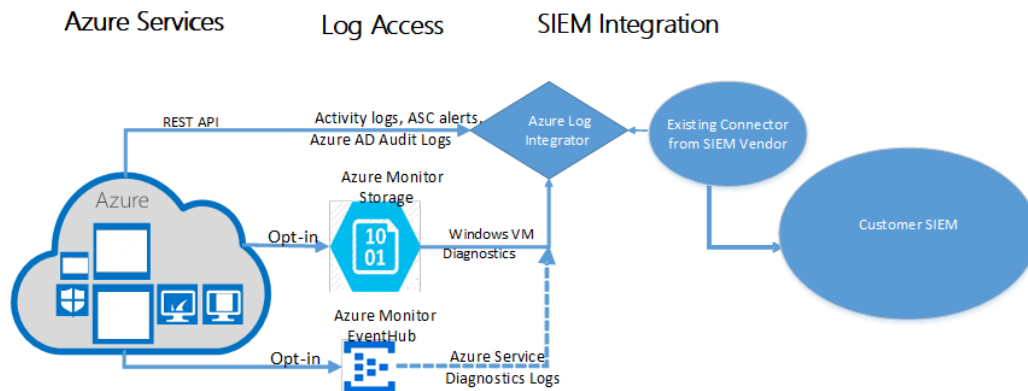


Figure 3-19: Azure Log Integration

Configure

Configure and automate operational tasks across your hybrid cloud environment such as collect inventory, track changes and configure desired state as well as manage patching. Automating operational tasks makes it possible for you to run at scale while minimizing human errors. It is also required to support modern development approaches for which you have need of an automated CI/CD pipeline.

Azure Automation

Azure Automation provides you the ability to automate frequent, time-consuming, and error-prone cloud management tasks. This automation helps you focus on work that adds business value. By reducing errors and boosting efficiency, it also helps to lower your operational costs. You can integrate Azure services and other public systems that are required in deploying, configuring, and managing your end-to-end processes. With the service, you can [author runbooks](#) graphically, in PowerShell or Python. By using a hybrid Runbook worker, you can unify management by orchestrating across on-premises environments. [Webhooks](#) provide a way to fulfill requests and ensure continuous delivery and operations by triggering automation from ITSM, DevOps, and monitoring systems.

Configuration management

Azure Automation [Desired State Configuration](#) is a cloud-based solution for PowerShell DSC that provides services required for enterprise environments. Manage your Desired State Configuration resources in Azure Automation and apply configurations to virtual or physical machines from a Desired State Configuration Pull Server in the Azure cloud. It provides rich reports that inform you of important events such as when nodes have deviated from their assigned configuration. You can monitor and automatically update machine configuration across physical and virtual machines, Windows or Linux, in the cloud or on-premises.

You can get inventory about in-guest resources for visibility into installed applications and other configuration items. Rich reporting and search capabilities are available to quickly find detailed information to help understand what is configured within the OS. You can track changes across services, daemons, software, registry, and files to quickly identify what might be causing issues. Additionally, Desired State Configuration can help you to diagnose and alert when unwanted changes occur in your environment.

Update management

You can update Windows and Linux systems across hybrid environments by using Azure Automation. You get visibility of update compliance across Azure, on-premises, and other clouds. You can create schedule deployments to orchestrate the installation of updates within a defined maintenance window. If an update should not be installed on a machine, you can exclude those updates from a deployment.

Chef Automate is an Azure Marketplace offering that gives you the capabilities you need to build, deploy, and manage your applications and infrastructure. Use the Chef Automate platform to package and test your applications, provision and update your infrastructure, and manage it all with compliance and security checks and dashboards that give you visibility into your entire stack (see [Automating Azure virtual machine deployment with Chef](#)).

Puppet Enterprise is another Azure Marketplace offering, with which you can automate the entire life cycle of your Azure infrastructure, from initial provisioning through application deployment. The Azure modules provision, configure, and manage Azure resources. The latest autogenerated [Azure ARM module](#) supports more than 220 existing Resource Manager resources and services, and as new ones are added, the module will be updated to support them, as well (see [Deploying Puppet Enterprise in Microsoft Azure](#)).

Update Management solution

The Update Management solution in Azure automation gives you the means to manage OS updates for your Windows and Linux computers deployed in Azure, on-premises environments, or other cloud providers. You can quickly assess the status of available updates on all agent computers and manage the process of installing required updates for servers.

You can set up Update Management for VMs directly from your Azure Automation account. To learn how to turn on Update Management for VMs from your Automation account, see [Manage updates for](#)

[multiple virtual machines](#). You can also set up Update Management for a single VM from the VM page in the Azure portal. This scenario is available to both [Linux](#) and [Windows](#) VMs.

Computers managed by update management use the following configurations for performing assessment and update deployments:

- Microsoft Monitoring agent for Windows or Linux
- PowerShell Desired State Configuration for Linux
- Automation Hybrid Runbook Worker
- Microsoft Update or Windows Server Update Services for Windows computers

Figure 3-20 shows a conceptual view of the behavior and data flow with how the solution assesses and applies security updates to all connected Windows Server and Linux computers in a workspace.

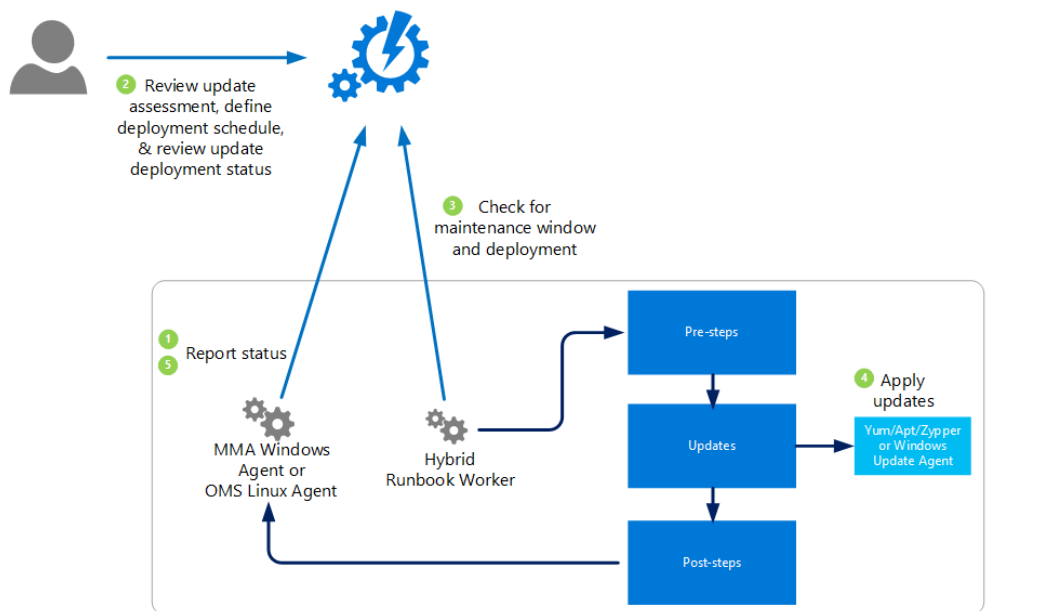


Figure 3-20: Update Management

Govern

IT Governance ensures that your organization is able to achieve its goals through an effective, secure and efficient use of IT. It does this by creating clarity between your business goals and IT projects.

Policy Management

Azure Policy is a service in Azure that you use to create, assign and, manage policies. These policies enforce different rules and effects over your resources so that those resources remain compliant with your corporate standards and SLAs. Azure Policy does this by running evaluations of your resources and scanning for those that are not in compliance with the policies you have created. For example, you can have a policy to allow only a certain SKU size of VMs in your environment. When this policy has been implemented, it will then be evaluated when creating and updating resources as well as over your already existing resources.

Policy definition

A policy definition has conditions under which it is enforced. And it has an accompanying effect that takes place if the conditions are met. There are some predefined policies in Azure that you can immediately use, but you can also create your own policies.

Policy assignment

A policy assignment is a policy definition that has been assigned to take place within a specific scope. The term scope refers to all the resource groups, subscriptions, or management groups to which the policy definition is assigned. Policy assignments are inherited by all child resources. This means that if a policy is applied to a resource group, it is applied to all the resources in that resource group. However, you can exclude a subscope from the policy assignment.

Policy parameters

Policy parameters help simplify your policy management by reducing the number of policy definitions that you must create. You can define parameters when creating a policy definition to make it more generic. Then, you can reuse that policy definition for different scenarios. You do so by passing in different values when assigning the policy definition. For example, specifying one set of locations for a subscription.

Initiatives

An initiative definition is a set of policy definitions that are tailored toward achieving a singular overarching goal. By applying an initiative to a scope, all policies within your initiative take effect.

Recommendations

While you're creating and managing policy definitions and assignments, here's some advice that we recommend you follow and tips to keep in mind:

- If you are creating policy definitions in your environment, we recommend starting with an audit effect, as opposed to a deny effect, to keep track of the impact of your policy definition on the resources in your environment. If you have scripts already in place to automatically scale up your applications, setting a deny effect might hinder those automation tasks you already have in place.
- It is important to keep organizational hierarchies in mind when creating definitions and assignments. We recommend creating definitions at a higher level—for example, at the management group or subscription level—and assigning at the next child level. For example, if you create a policy definition at the management group level, a policy assignment of that definition can be scoped down to a subscription level within that management group.
- We recommend always using initiative definitions instead of policy definitions, even if you have only one policy in mind. For example, if you have a policy definition—policyDefA—and you create it under the initiative definition—initiativeDefC—if you decide to create another policy definition later for policyDefB with goals similar to that of policyDefA, you can add it under initiativeDefC and track them better that way.
- Keep in mind that after you have created an initiative assignment from an initiative definition, any new policy definitions added to the initiative definition automatically roll under the initiative assignment(s) under that initiative definition. However, if there's a new parameter introduced to the new policy definition, you need to update the initiative definition and assignments by editing the initiative definition or assignment.
- After an initiative assignment is triggered, all policies within the initiative will be triggered, as well. However, if you needed to execute a policy individually, it is better to not include it in an initiative.

Topic	Resource
Overview of Azure Policy	https://docs.microsoft.com/azure/azure-policy/azure-policy-introduction

Cost Management

Cost Management helps you to best utilize and manage your cloud resources. It gives you the ability to track cloud usage and expenditures for your Azure resources and other cloud providers, including Amazon Web Services and Google.

Monitor usage and spending

Monitoring your usage and spending is critically important for cloud infrastructures because organizations pay for the resources they consume over time. When usage exceeds agreement thresholds, unexpected cost overages can quickly occur. A few important factors can make ad hoc monitoring difficult. First, projecting costs based on average usage assumes that your consumption remains consistent over a given billing period. Second, when costs are near or exceed your budget, it's important that you get notifications proactively to adjust your spending. And, cloud service providers might not offer cost projection versus thresholds or period-to-period comparison reports.

Reports help you to monitor spending to analyze and track cloud usage, costs, and trends. Using Over Time reports, you can detect anomalies that differ from normal trends. Inefficiencies in your cloud deployment are visible in optimization reports. You can also notice inefficiencies in cost analysis reports.

Manage costs

Historical data can help manage costs when you analyze usage and costs over time to identify trends. Trends are then used to forecast future spending. Cost Management also includes useful projected cost reports.

Cost allocation manages costs by analyzing your costs based on your tagging policy. You can use tags on your custom accounts, resources, and entities to refine cost allocation. Category Manager organizes your tags to help provide additional governance. And, you use cost allocation for showback/chargeback to show resource utilization and associated costs to influence consumption behaviors or charge internal customers.

Access control helps manage costs by ensuring that users and teams access only the cost management data that they need. You use entity structure, user management, and scheduled reports with recipient lists to assign access.

Alerting helps manage costs by notifying you automatically when unusual spending or overspending occurs. Alerts can also notify other stakeholders automatically for spending anomalies and overspending risks. Various reports support alerts based on budget and cost thresholds.

Improve efficiency

You can determine optimal VM usage and identify idle VMs or remove idle VMs and unattached disks with Cost Management. Using information in Sizing Optimization and Inefficiency reports, you can create a plan to down-size or remove idle VMs.

Topic	Resource
Cost Management Overview	https://docs.microsoft.com/azure/cost-management/overview

Managed services for standard and business applications

Managed services are not a new business model. For more than 20 years, large enterprises have relied on service providers (internal IT departments or external service providers) to manage their IT assets. Managed-service providers have been managing their customers' workloads—either in their own datacenters or those operated by their customers. This section focuses on internal IT organizations that act as managed service providers. However, the basic concepts remain the same for external managed service providers.

The cloud requires a new method of management because of its focus on scale, elasticity, and automation. The cloud represents a paradigm shift in the way that we think about embracing IT. DevOps has completely changed the way applications are developed and maintained. The hyperscale nature of the cloud provides a completely new meaning to scalability, elasticity, and resiliency, and has redefined how applications are designed and delivered. The pay-as-you-go model provides a fail-fast, Agile method of app development. Because of the cloud, IT organizations require a new way to think about data governance and security.

Managed-service providers for cloud services are organizations that help their customers transition to this paradigm shift in technology by guiding the customers in various aspects of the cloud journey. Successful Azure managed-service providers differentiate themselves by building a practice around DevOps, automation, and cloud-native application design. They use the best Azure features while designing solutions—be it IaaS, PaaS, or SaaS offerings—to meet their customers' business requirements. Essentially, they act as a one-stop shop for their customers by providing a common support, provisioning, and billing experience—all with a flexible pay-as-you-go business model.

For managed service providers, automation and orchestration are extremely important functions to a successful practice. The ability to automate routine tasks makes it possible for you to lower your delivery costs and offer superior SLAs, driving a virtuous cycle of efficiency and repeat business. Automation is the key to creating the right balance between cost, reliability, speed, and time to market.

Managed services are centrally managed along the entire application life cycle and include all operational components such as Backup, monitoring, updating, and so on and are managed according to the organization's standards. They are typically offered in different pricing tiers, such as those shown in Figure 3-21.

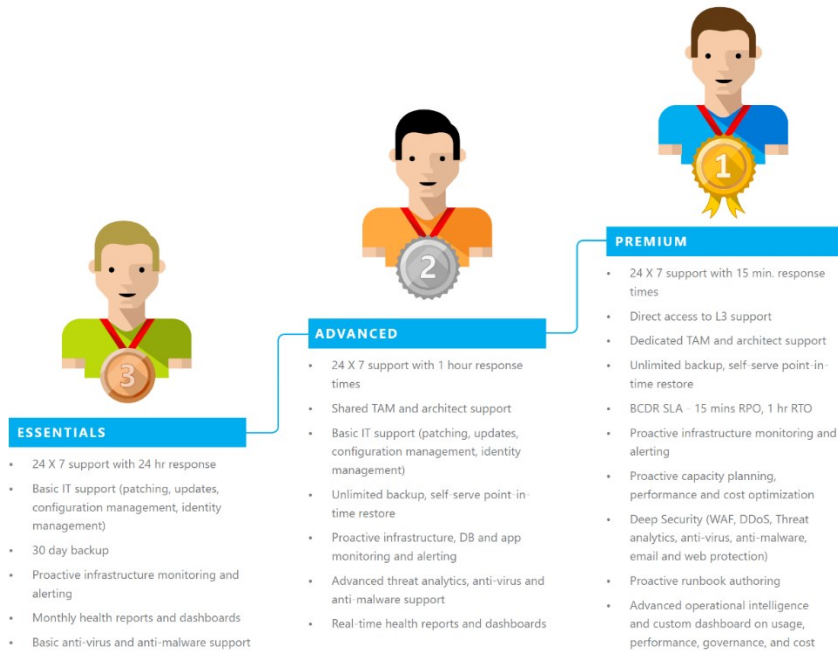


Figure 3-21: Pricing tiers for managed services

Offering managed services

IT organizations require a comprehensive integrated solution that gives their consumers the complete visibility and control of their cloud resources while letting them fully exploit Azure's extensive capabilities and empowering agility. Therefore, providers can take advantage of different frameworks to build such an integrated solution. This section doesn't focus on any specific framework; rather, it outlines only the required capabilities and the conceptual layers of such a solution.

The required logical layers are shown in Figure 3-22.

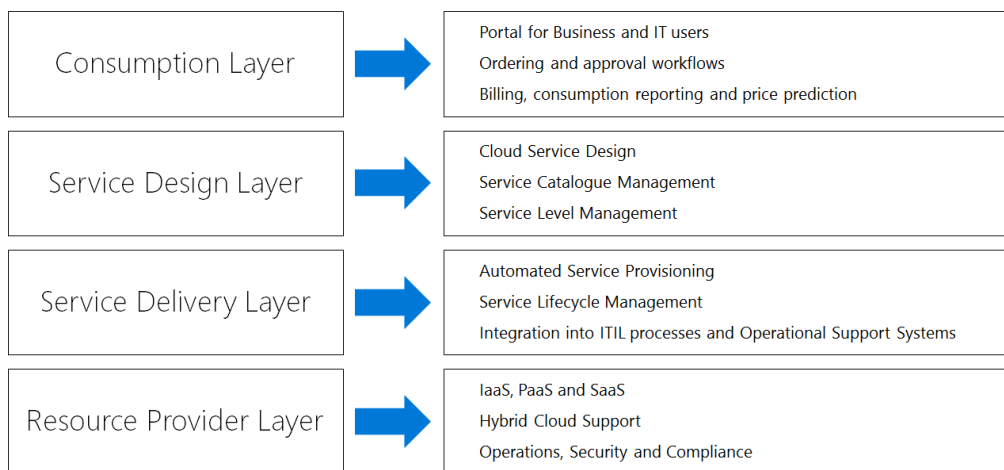


Figure 3-22: Conceptual model for managed services

The consumption layer presents a service catalog to the consumer, with various configuration options (Application, Compute, RAM, Storage, Location, Network, SLA, etc.), including pricing information. Approval workflows need to be available to control costs and to govern the consumption of the services.

The service layer is used for managing the service catalog, defining the SLAs (Availability, Performance, Time to React, etc.) that are associated with the provided services and to design the service itself. A service consists not only of a VM with an operating system. Enterprises require managed services for operating systems, databases, middleware products, web servers, or complex business applications (e.g., an ERP solution) that are based on multiples of these components. The service layer must support an efficient way to design new services based on standard products as well as the ability to orchestrate multiple products and services to a new complex business service.

The services must include availability monitoring, backup and restore processes, antivirus protection, security, and compliance monitoring and must be managed according to service management standards.

The service provisioning as well as the integration with the operational support systems and the IT service management tooling is part of the service delivery layer. Those processes are ideally automated but could also be semi-automated or run manually with a defined service level for delivering the service to the consumer.

Finally, there is a Resource Provider Layer with IaaS-, PaaS-, and SaaS-based services that is used as foundation to create managed services. A large portion of enterprise IT is still focused on IaaS, but the obvious benefits of PaaS and SaaS are leading to a higher adoption of those services.

We observe that customers are sometimes aiming for a multivendor strategy on the resource provider layer called *multicloud support* or *cloud brokerage*. But there are a lot of caveats with such an approach:

- Each cloud provider has its own standards and technology. Adopting a multicloud approach will actually reduce the agility and the capabilities that customers get from the cloud services they are paying for. Higher-level services like PaaS and SaaS differ heavily between the various providers, and on IaaS, there is also a significant discrepancy in regard to network, security, and resilience, operations and life cycle management. Using the lowest common denominator between the clouds has significant disadvantages for the business.
- The speed of innovation in Azure as well as the frequency of releasing new services is very challenging for any integration layer. Business departments want to benefit from the new capabilities but the cloud brokering platforms aren't supporting it in a timely manner.
- The complexity and lock-in that is introduced with such an integration layer is enormous.

IT organizations should rate for themselves the pros and cons of multicloud support.

Consuming managed services

The user-self-service capability is an essential characteristic of cloud computing, and it must be present in any implementation. The intent is to permit users to approach a self-service capability and be presented with options available for provisioning. The capability can be basic (such as provisioning of a VM with a predefined configuration), more advanced (such as allowing configuration options to the base configuration), or complex (such as implementing a platform capability or service).

The self-service capability is a critical business driver for customers to become more agile in responding to business needs with IT capabilities that align and conform to their internal business and IT requirements. The interface provided by the IT organization should be abstracted to a well-defined, simple, and approved set of service options. The options should be presented as a menu in a portal. Customers should be able to select these services from the catalog, start the provisioning process, and be notified upon completion. Customers should be charged only for the services they actually used.

The challenge is to find the right balance between the required configuration options to fulfil the business needs and the complexity to implement those options and provision the services

accordingly. Adding all configuration options that Azure provides for a service to the service catalog doesn't make sense. It would end up in re-creating the Azure portal which isn't achievable for any IT organization. A common approach is to add the most important parameters, that allow a basic automated provisioning of the service from the service catalog. Detailed configuration settings could be requested by the customer in separate service request (ideally via the same portal) and could be performed by IT staff within a defined service level.

Provisioning managed services

The infrastructure for an application is typically made up of many components—maybe a VM, storage and virtual network, or a web app, database, database server, and probably third-party services from the Azure Marketplace or custom solutions that have been developed by the business. These components shouldn't be considered as separate entities, instead you should view them as related and interdependent parts of a single entity. IT departments should use Azure Resource Manager combined with DSC, Puppet, or Chef to deploy, manage, and monitor them as a group. Using Resource Manager, you can deploy, update, or delete all the resources of an application in a single, coordinated operation. Templates could be used for deployment, and that template can work for different environments such as testing, staging, and production. The template includes the infrastructure for the application, how to configure that infrastructure, and how to publish the application code to that infrastructure. Resource Manager also provides security, auditing, and tagging features to help service providers to manage the resources after deployment.

Metering consumption

Consumers require the ability to get an accurately predicted price before deploying an application. As they move from a capital expenditure (Capex) to an operating expenditure (Opex) model, they also need the ability to do showback versus chargeback analysis as well as provide mode fidelity in estimation and billing, especially for large deployments.

The Azure Resource Usage and Rate Card APIs (Figure 3-23) address these needs by providing new insights into the consumption of Azure resources. With the Azure Usage API, you can programmatically pull in usage data to gain insights into the consumption. The granularity (hourly usage information) and resource metadata information available through the API provides the necessary dataset to support flexible Showback or Chargeback models.

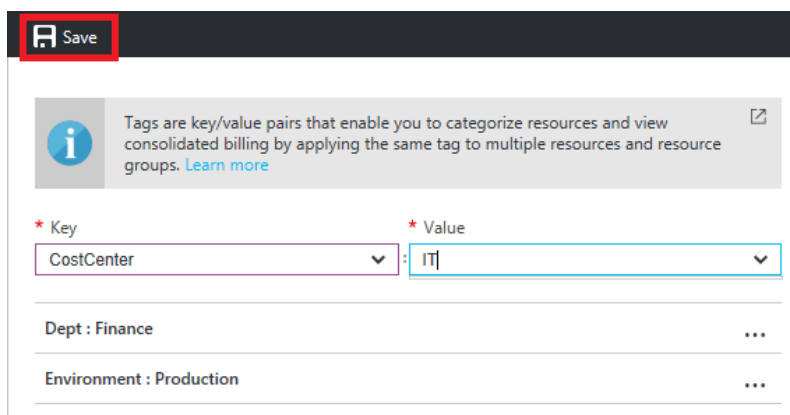


Figure 3-23: Metering consumption

The data available through the Azure Usage API includes not only consumption information, but also resource metadata including any tags associated with it, as illustrated in Figure 3-24. Tags provide an easy way to [organize application resources](#), but to be effective, you must ensure the following:

- Tags are correctly applied to the application resources at provisioning time

- Tags are properly used on the Showback/Chargeback process to tie the usage to the organization's account structure.

Daily Usage						Tags
Usage Date	Meter Category	Unit	Consume	Resource Gr	Instance Id	
5/14/2015	"Virtual Machines"	"Hours"	3.999984	"computeRG"	"virtualMachines/catalogVM"	{ "costCenter": "finance", "env": "prod" }
5/14/2015	"Virtual Machines"	"Hours"	3.999984	"businessRG"	"virtualMachines/dataVM"	{ "costCenter": "hr", "env": "test" }

Figure 3-24: Usage report

Table 3-7 lists some additional resources.

Table 3-7: Backup platform services

Topic	Resource
Understand your Azure bill	https://docs.microsoft.com/azure/billing/billing-understand-your-bill
Use Azure Billing APIs to programmatically get insight into your Azure usage	https://docs.microsoft.com/azure/billing/billing-usage-rate-card-overview

Billing and price prediction

A proper price prediction for a potentially consumed service from the service catalog is a common demand. Managers require this information to approve large cloud application deployments. Price prediction requires detailed knowledge about the Azure elements that a service consist of (defined at design time) along with estimated pricing information for each.

The Azure Resource RateCard API delivers a list of available Azure resources and pricing information. The API provides Azure offer-level rate information versus subscription-level. The caller of this API must pass in the offer information to get resource details and rates. As Enterprise Agreement (EA) offers have customized rates per enrollment, the API is unable to provide the EA rates at this time.

For EA scenarios, you can use another API for Enterprise Agreement that allows usage access price sheet and other billing information in CSV and JSON format.

Service management

IT service management is not obsolete when you use cloud technologies—on the contrary, effective management of cloud resources and service integration becomes more important than ever. However, the role of IT service management does change, especially with product teams delivering and operating their products at a faster pace.

The classic IT Infrastructure Library (ITIL) service life cycle shown in Figure 4-1 demonstrates how a continuous cycle of service improvement can be centrally coordinated and organized for an organization. The challenge and the opportunity in today's cloud world is to effectively manage, together with the individual product teams, an organization's workloads to utilize the possibilities for optimization now available through hybrid and public cloud infrastructures.

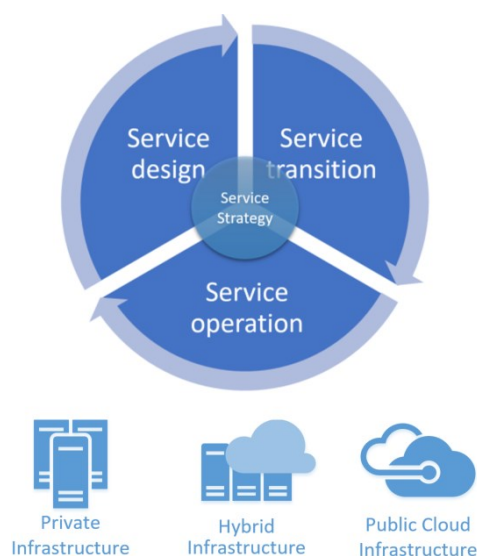


Figure 4-1: ITIL service life cycle with private, hybrid, and public cloud infrastructure options

In the cloud and hybrid world, the diversity of services and operations means that there is not one service management for all; each workload must be evaluated individually with the team responsible for the workload. An agreement, or contract of sorts, should be defined per workload that defines which organizational policies and best practices will be used from IT, and which responsibilities are overtaken by the product team to best serve the business and users as well as ensure optimal operations. It is very important to have a clear agreement between IT and the product teams before you first deploy the service.

Figure 4-2 shows the service management areas divided between the IT and the product teams, because in every case, both organizations will be involved to different degrees. Also important is the flow of information regarding organizational policies and best practices from IT to the product teams, and in the other direction, the overtaking of some responsibilities by the product teams from IT.

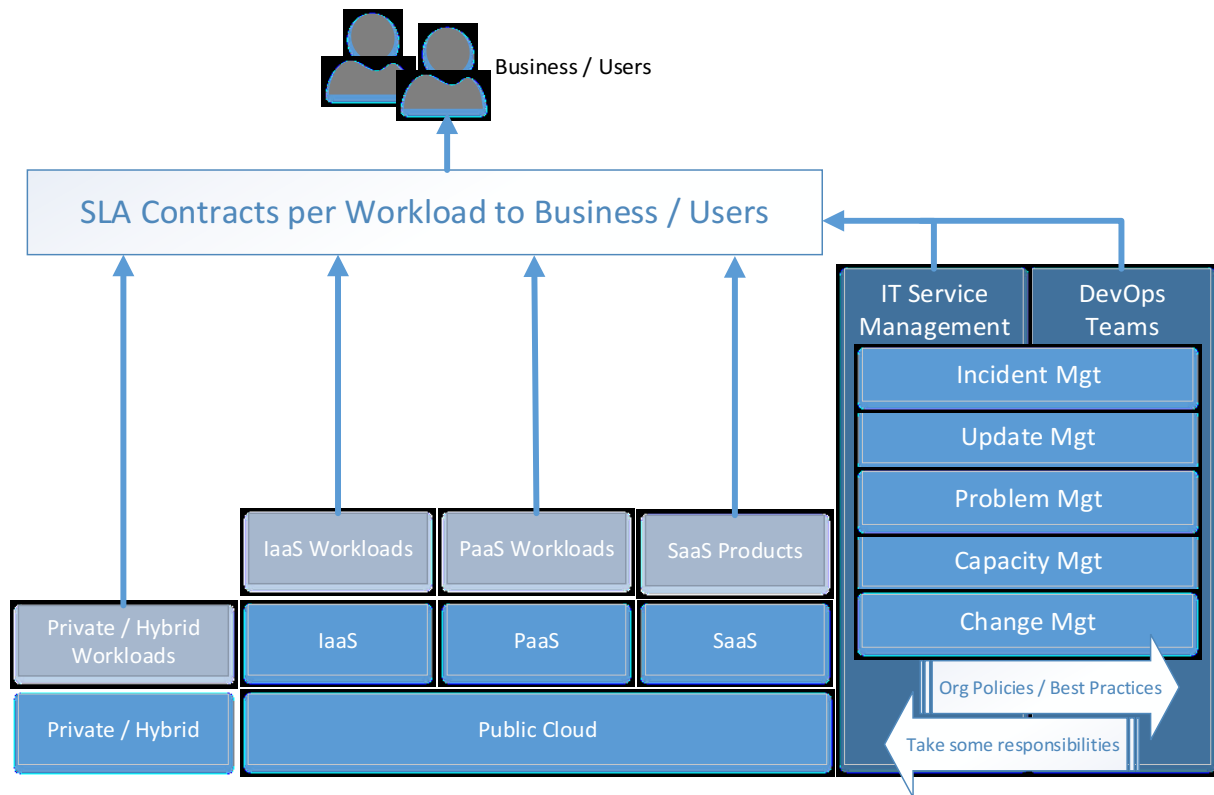


Figure 4-2: Modern service management with an agreed approach between IT and product teams

Incident management

The goal of incident management is, as its name implies, to restore functionality as soon as possible to the business or to respond as soon as possible to a service request in the event of unstable performance. In both cases the goal is that operations continue at the agreed level without interruptions.

The integration of incident response processes across private, hybrid, and cloud infrastructures is one of the first challenges of cloud readiness.

Figure 4-3 shows not only how monitoring is simplified in the public cloud (regional Microsoft Azure status information replaces individual hardware and network monitoring), but also the challenge of unifying monitoring across the private, hybrid, and public infrastructures.

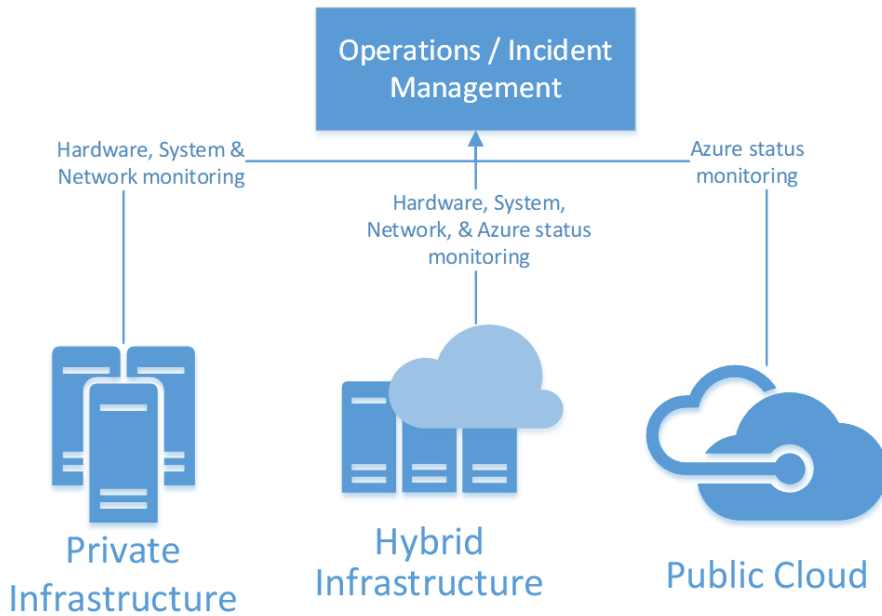


Figure 4-3: Monitoring across private, hybrid, and public cloud infrastructure

Incidents with apps deployed in the public cloud usually occur in the following scenarios (see also Figure 4-4):

- An Azure status or service health information raises an alert preemptively, which could affect service functionality, which can result in the creation of an incident ticket in the incident management system.
- A problem is encountered while a cloud app or service is being used, which initiates the opening of an incident in the corporate incident management system.
- The cloud apps and services are running normally, but a problem is encountered during the deployment of an update or the activation of a feature flag that had been previously turned off.

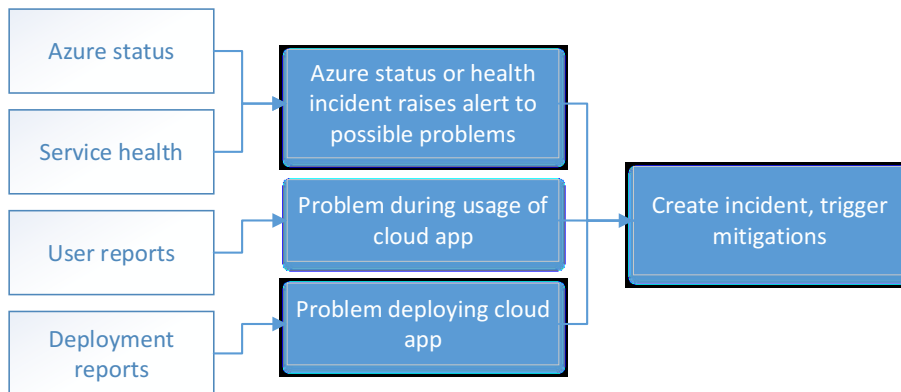


Figure 4-4: Decision process for incident response

In the same way that hardware and network monitoring information can prompt alerts and operations processes for private infrastructure, we recommend that you integrate the Azure status and monitoring information into the IT organization in the same way.

Azure status

The global and regional status of all Azure platform services and datacenters is available at the [Azure Status](#) website, shown in Figure 4-5.

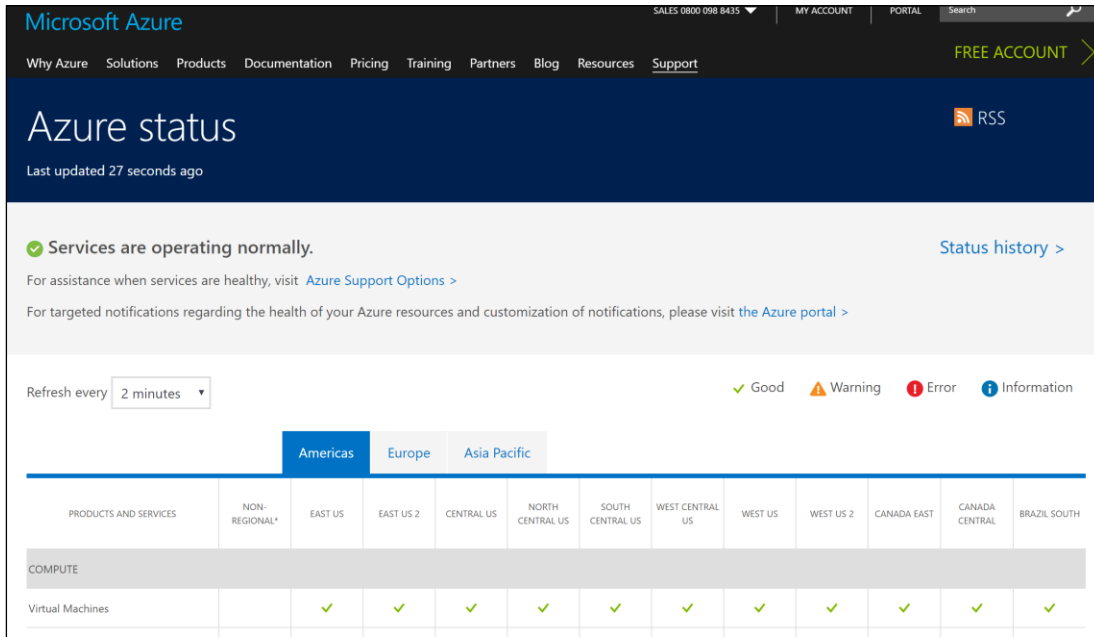


Figure 4-5: The Azure Status site including RSS feed link and status history

Any alerts or current incidents that might affect customers are published at the top of the page. Users can subscribe to an RSS feed to get updates pushed immediately. Customers are responsible for monitoring this information and taking preemptive action if a regional incident could be of importance to their deployments.

A detailed history of all status alerts is also provided with the resolution time and incident details. Table 4-1 provides a link where you can find additional information about Azure status.

Table 4-1: Azure status

Topic	Comment
Azure status	https://azure.microsoft.com/status/

Service health

Just as important as the global Azure status is the status and health of the user's individual services and deployments. This information is available in the Azure portal, both as a dashboard (Figure 4-6) and historical health data for individual Azure resources such as Virtual Machines and SQL Database (click Help + Support > Resource Health). Resource health is organized by resource group so that you can get an overview of the health status for a logical group of Azure services.



Figure 4-6: Resource health dashboard in the Azure portal

Historical data of the resource health is automatically saved for 14 days, as illustrated in Figure 4-7.

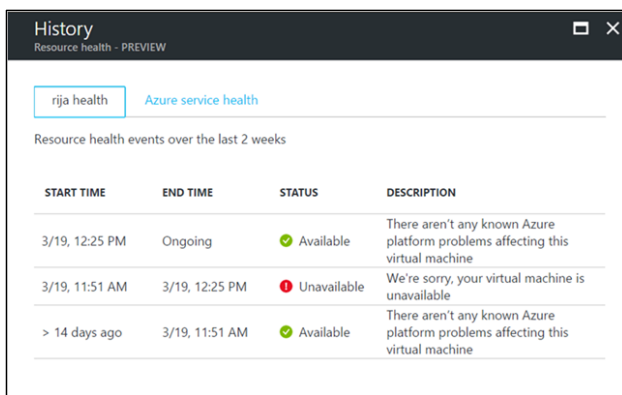


Figure 4-7: Historical resource health data in the Azure portal

More info To read more about Service Health and Service Health Alerts, see the section “Monitoring” in Chapter 3.

Table 4-2 provides links to where you can find additional information.

Table 4-2: Service health

Topic	Comment
Resource health overview	https://docs.microsoft.com/azure/resource-health/resource-health-overview
Reduce troubleshooting with Azure Resource Health	https://azure.microsoft.com/blog/reduce-troubleshooting-time-with-azure-resource-health/
How to use the Resource Health API	https://blogs.msdn.microsoft.com/premier_developer/2017/04/06/how-to-use-azure-resource-health-api-to-gain-visibility-into-the-health-of-a-vm-web-app-or-sql-database/
Azure Insights Alerts Portal	https://docs.microsoft.com/azure/monitoring-and-diagnostics/insights-alerts-portal
Integrate Azure alerts with PagerDuty, OpsGenie, VictorOps	https://azure.microsoft.com/blog/webhooks-for-azure-alerts/?v=17.23h

IT Service Management integration

IT Service Management (ITSM) integration can drastically help in simplifying the monitoring landscape, especially when both on-premises and cloud resources are in use. The Azure services in use for this integration are Azure Log Analytics, Azure Monitor, Azure Security Center, Policy Management, Update Management, and Automation, which offers a comprehensive operations suite across both on-premises and cloud resources and offers the integration into ITSM systems to manage work items across products and services.

Azure Log Analytics is described in more detail in Chapter 3, in the section “Log Analytics.” The feature ITSM Connector (currently in preview) provides integration of work items and incidents with these ITSM solutions:

- [System Center Service Manager](#)
- [ServiceNow](#)
- [Provance](#)
- [Cherwell](#)

Table 4-3 provides a link where you can find more information.

Table 4-3: ITSM Integration

Topic	Comment
Centrally manage ITSM work items using IT Service Manager Connector (Preview)	https://docs.microsoft.com/azure/log-analytics/log-analytics-itsmc-connections

Security incidents

An advantage of public cloud Azure deployments is that they benefit from the automatic outer perimeter protection that Microsoft provides to the entire Azure infrastructure. This means that antimalware programs, distributed denial of service (DDoS) protection, and advanced threat analytics are active and protect every Azure workload. You can use the Azure DDoS Standard service to customize the DDoS protection your application receives.

Microsoft has decades of experience in the world of internet security and protecting user’s most valuable data; the company applies that experience and associated processes to every Azure subscription and deployment. This protection is an advantage of Azure public cloud deployments and helps IT, product, and security teams focus their attention on the internal security of deployed workloads. Figure 4-8 shows how the potential attack area is reduced in cloud deployments, visualized by the white boxes. The gray boxes on the right side are no longer vulnerable because of the built-in protections of the Azure platform.

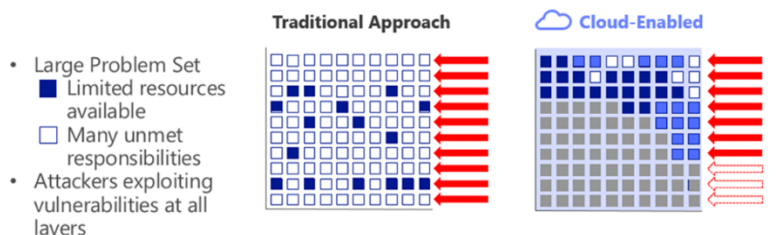


Figure 4-8: The advantage of utilizing Azure infrastructure protection

The Microsoft Security Response Center (MSRC) receives every potential security escalation. Figure 4-9 depicts the process it follows for managing both security and availability incidents in Azure. This process provides the response path for every reported incident, with the clear goal of restoring normal operations as quickly and smoothly as possible.

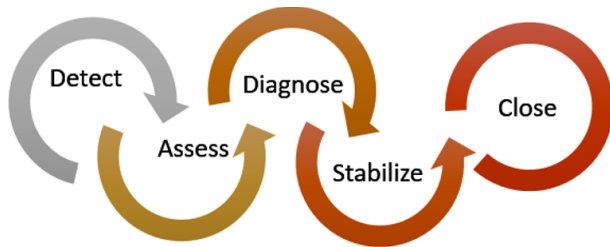


Figure 4-9: Security incident response life cycle at Microsoft

Customers are responsible for monitoring and detecting security threats and incidents in their own software deployments, but they can take advantage of the included security integration through Azure Security Center to further detect security incidents.

Log Analytics also provides security incident integration to your on-premises analytics/Security Information and Event Management (SIEM) system through the Azure log integration functionality, as illustrated on Figure 4-10.

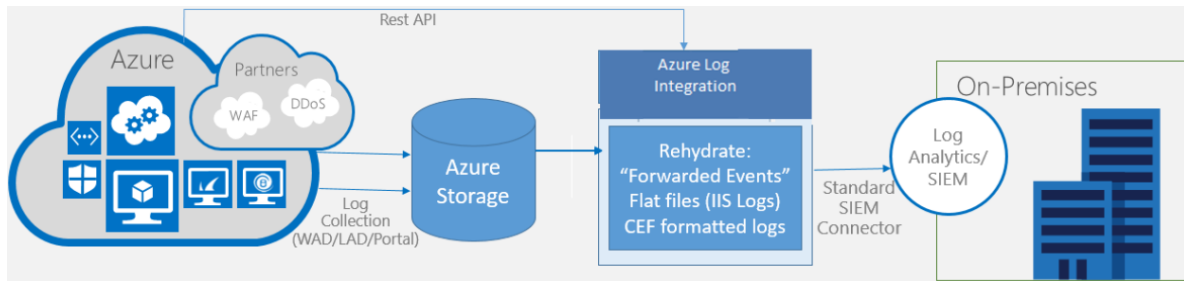


Figure 4-10: The flow of Azure security information to a local SIEM system

Integrating Azure Log information into your Log Analytics/SIEM system has the advantage of collecting all logs in one place and helps integrate Azure resources with existing infrastructure and alert configurations.

Azure Security Log Integration has partner integrations out of the box with these systems:

- Splunk
- ELK
- HP ArcSight
- IBM QRadar

The Security Center is available in the Azure portal and gives security guidance based on the current resources deployed in the Azure subscription. It is a best practice to periodically review this information together with IT security experts and the product team.

The security of all Microsoft and Azure cloud products is described at the Microsoft Trust Center. While you're there, if you have a registration, you can view and download copies of compliance and regulation certifications. Table 4-4 provides links to additional information.

Table 4-4: Security incidents

Topic	Comment
Azure Log integration overview	https://docs.microsoft.com/azure/security/security-azure-log-integration-overview
Azure Log SIEM configuration steps	https://blogs.msdn.microsoft.com/azuresecurity/2016/08/23/azure-log-siem-configuration-steps/
Microsoft whitepaper Security Incident Response	http://aka.ms/SecurityResponsepaper
Azure Security Center	https://docs.microsoft.com/azure/security-center/security-center-intro
Microsoft Trust Center	https://www.microsoft.com/TrustCenter
Azure DDoS Protection Standard overview	https://docs.microsoft.com/en-us/azure/virtual-network/ddos-protection-overview

Microsoft support

To ensure that you can remedy incidents with as little delay as possible, we strongly recommend that you include Microsoft support in the clarification process. Microsoft support engineers bring the experience from similar deployments, and they can help clarify root causes early to avoid costly searching. Microsoft support can also help with security and architecture reviews at an early stage in the service life cycle to help avoid incidents from occurring in the first place.

Every Azure subscription has the option of adding support that is directly integrated in the Azure portal (Figure 4-11), which greatly simplifies opening and tracking tickets. You can create support tickets in the Azure portal, via the API, by telephone, or by email.

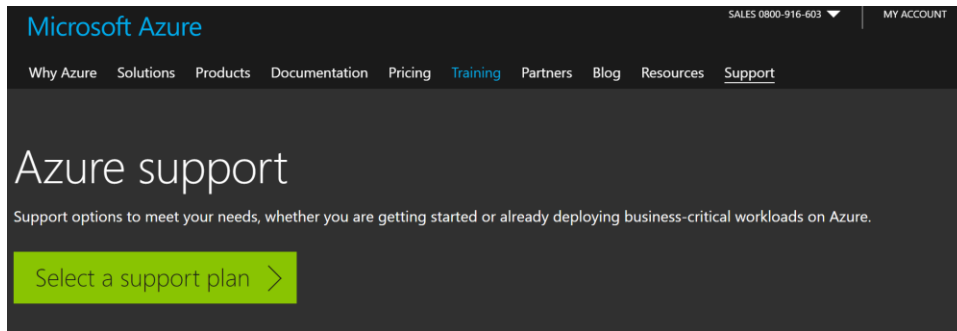


Figure 4-11: Microsoft support plan options for Azure

Furthermore, it is possible to automate and integrate Microsoft support through email or API calls, as illustrated in Figure 4-12; for example, into the corporate incident management system. This way, there is no danger of losing the connection between an incident and any support assistance that Microsoft can provide.

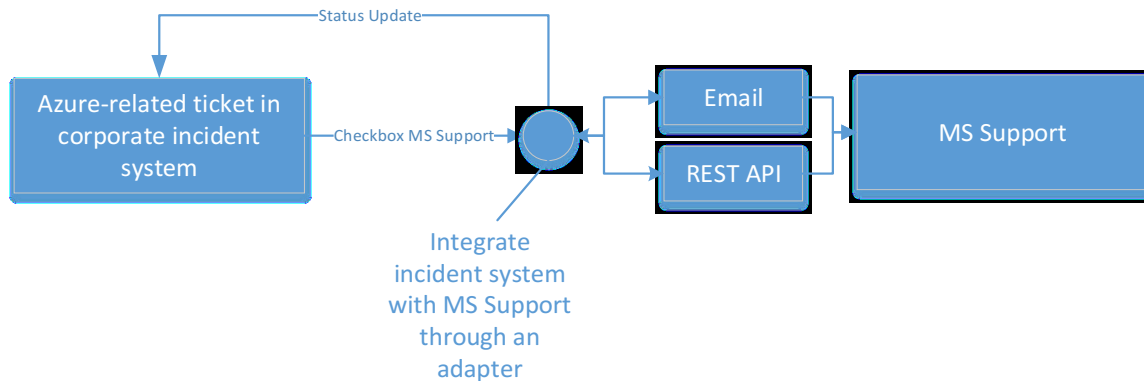


Figure 4-12: Integration between a corporate incident ticketing system and Microsoft support

Because Microsoft support is connected to a support contract, it is a governance decision as to who should be allowed to create support tickets. You can limit this to dedicated users in the Azure portal.

You can, for example, assign mission-critical apps to have direct access to request Microsoft support, but internal apps or those with a lower priority must go through an internal process to validate the support request before sending. Table 4-5 provides links to additional information.

Table 4-5: Microsoft support

Topic	Comment
Microsoft Support plans	https://azure.microsoft.com/support/options/
Support Requests Access Control	https://docs.microsoft.com/azure/azure-supportability/create-manage-support-requests-using-access-control

Problem management

Problem management means recording recurring incidents and problems of all types and implementing processes to guide solutions that can reduce or eliminate those problems in the future—of course, prioritized to the problems with the biggest business impact.

In the Scrum framework, this is referred to as a *Retrospective*, in which all stakeholders gather together after a sprint to collect change suggestions that can be implemented in the next sprint. The principle of continuous feedback is important. You need to continually observe what worked and what didn't and take action to improve recurring and fixable problems for the future.

DevOps and cloud operations can bring a new dynamic to continuous improvement. This happens because feedback and input come much faster—continuous deployment means that software is released much more often; thus, feedback and input for improvement come much more often, as well. Figure 4-13 illustrates the process.

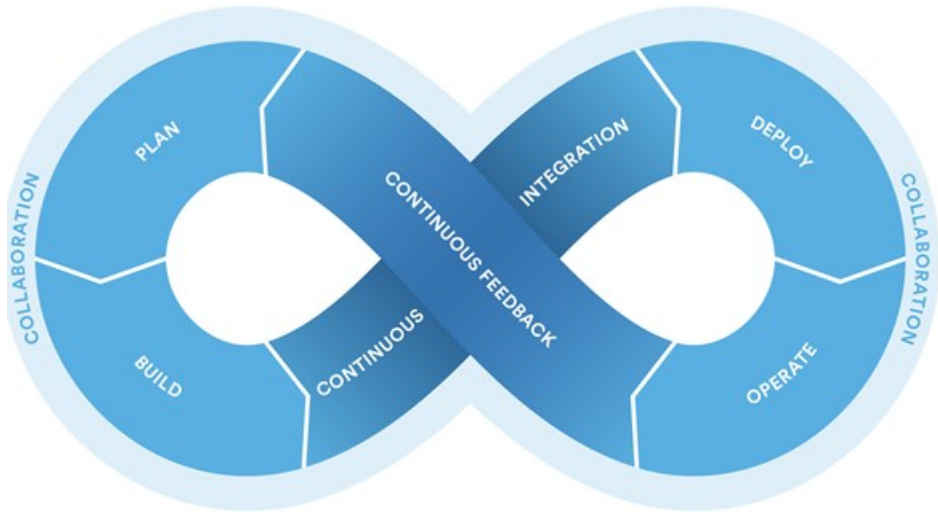


Figure 4-13: DevOps emphasizes collaboration and a continuous cycle of release and feedback

Azure feedback can be given on Twitter under ([#AzureSupport](https://twitter.com/AzureSupport)), through a Microsoft Support plan, or through the Azure Feedback portal. Table 4-6 provides links to additional information.

Table 4-6: Problem management

Topic	Comment
Microsoft support plans	https://azure.microsoft.com/support/options/
Azure support via Twitter	https://www.twitter.com/azuresupport
Azure feedback	https://feedback.azure.com/

Change management

Because continuous delivery is focused on the relatively short-term development and delivery cycle, it is very helpful to have a dedicated innovation and change track that thinks longer term.

This change and innovation track should take input from incidents, problems, vendor roadmaps, technology innovations, and other sources to plan the mid- to long-term technology and innovation strategy of the organization, so thinking in months and years instead of in weeks and sprints.

Figure 4-14 demonstrates how you should integrate factors that influence change—whether coming through problem management, service life cycles, or roadmaps—into an innovation cycle that drives change in an organization. This can help avoid (sometimes costly) surprises if important changes are not taken into account.

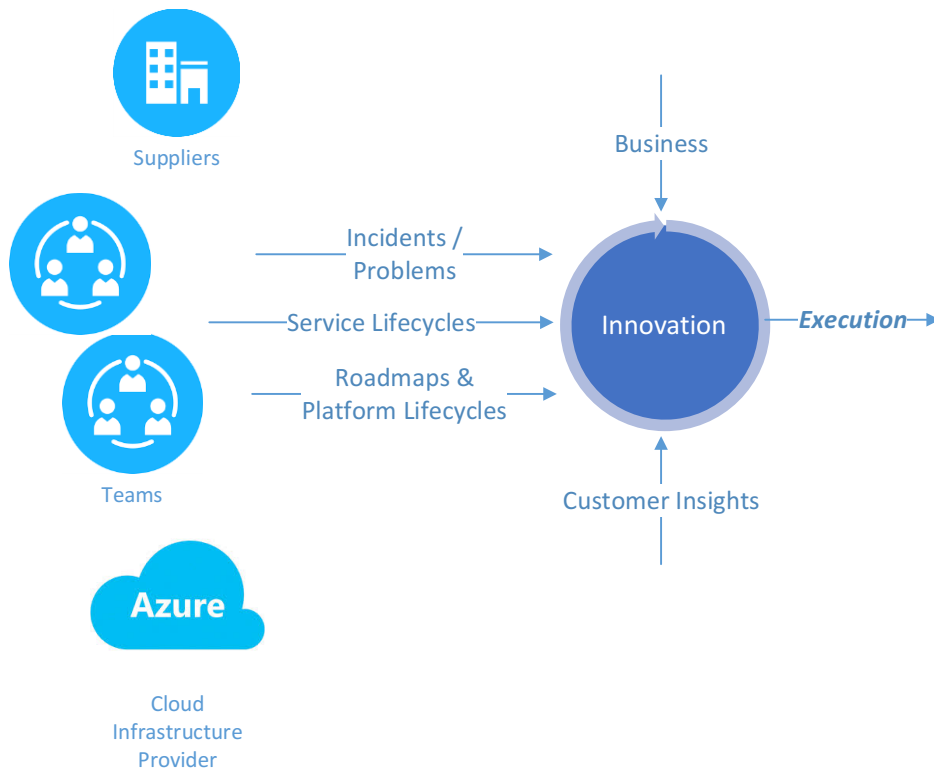


Figure 4-14: Input from suppliers, project teams, and cloud infrastructure providers, as well as business and customers to drive change and innovation

Azure platform change

Figure 4-14 is a living part of the Azure team, as well. Input is gathered from a variety of sources and fed into the Azure roadmap. Customers and users are also heavily involved in this process and are invited to give feedback and vote on features at the Azure Feedback website.

The results of the change cycle are posted on the Azure Roadmap website (Figure 4-15), which is available to the public. Here features are documented that are in development, preview, and release stages.

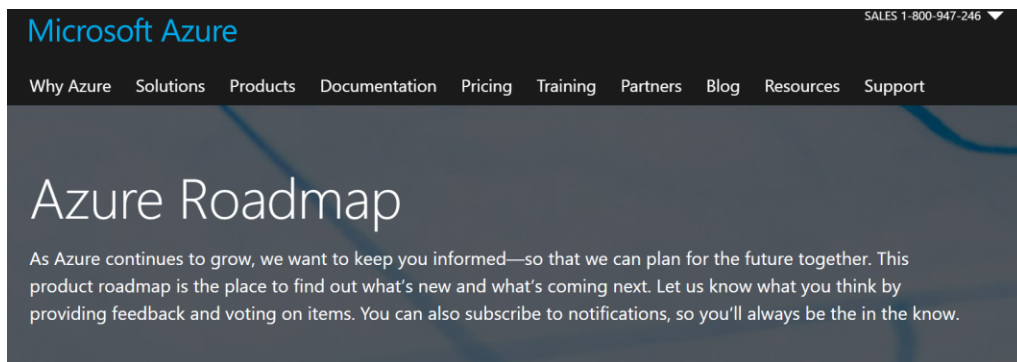


Figure 4-15: The Azure Roadmap website lists features in development, in preview, and in general availability

All updates and changes to the Azure platform, including new features and rollouts of existing features to new regions, are documented on the [Azure Updates](#) site, as shown in Figure 4-16.

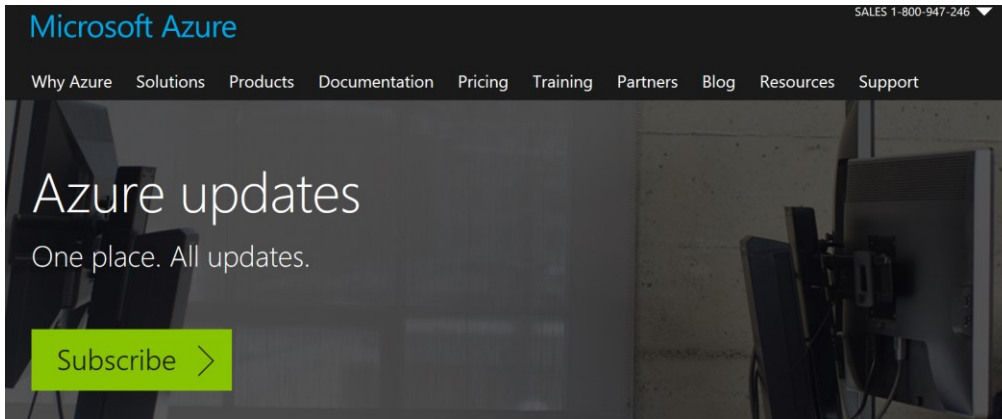


Figure 4-16: The Azure Updates website lists all new features and rollouts of existing features to new regions

As illustrated in Figure 4-17, this cycle continually drives the Azure platform forward and delivers a rapid pace of innovation and delivery for Azure customers and users.

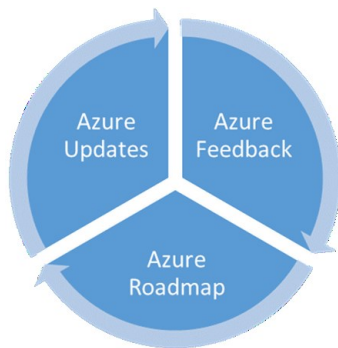


Figure 4-17: Input from customers through feedback drives the roadmap, which results in continuous updates

Despite this rapid pace of innovation, stability and reliability are key pillars of the Azure platform and are backed up by Service-Level Agreement (SLA) guarantees for every nonpreview released service.

The individual life cycles of the Azure platform services are documented on the Microsoft Lifecycle site.

For all paid and nonpreview Azure services, you are guaranteed at least a 12-month notification for a service cancellation without a replacement product. Additional support and migration assistance is available through the Microsoft Support service plans. Table 4-7 provides links where you can learn more.

Table 4-7: Platform change

Topic	Comment
Microsoft Support Life cycles	https://support.microsoft.com/lifecycle
Microsoft Support Plans	https://azure.microsoft.com/support/options/
All service SLAs	https://azure.microsoft.com/support/legal/sla/
Azure Roadmap	https://azure.microsoft.com/roadmap/
Azure Update	https://azure.microsoft.com/update/

Capacity management

The goal of IT capacity management is to ensure that IT resources are properly sized to meet current and future resource demands as well as to ensure that those resources are provisioned in a cost-effective manner. A major advantage of Azure cloud resources is that you can flexibly manage and dynamically scale the capacity as needed on a minute-by-minute basis, which allows for maximum cost efficiency compared to static on-premise resources.

This is nothing short of a revolution in capacity planning when compared to the lead and ordering times of provisioning servers and hosting in the past. There are basically no downsides to this flexibility, the cloud is ultimately delivering on the process of dynamically allocated compute resource pools that have been the dream of datacenter operators for decades. Finally, "right-sizing" is a reality.

You manage Azure capacity by using subscriptions and resource groups, with which you can set limits on spending (on a subscription level) and assign roles and teams (on both subscription and resource group level).

Because you can plan, assign, and change the resources on demand, capacity planning becomes a fluid topic that provides an organization with the ability to flexibly change and react and use resources as effectively as possible at any given time. Enterprise Agreement (EA) customers can define department spending limits per account in the EA portal, which helps to give an initial guidance to departments for specific projects. Figure 4-18 shows an overview.

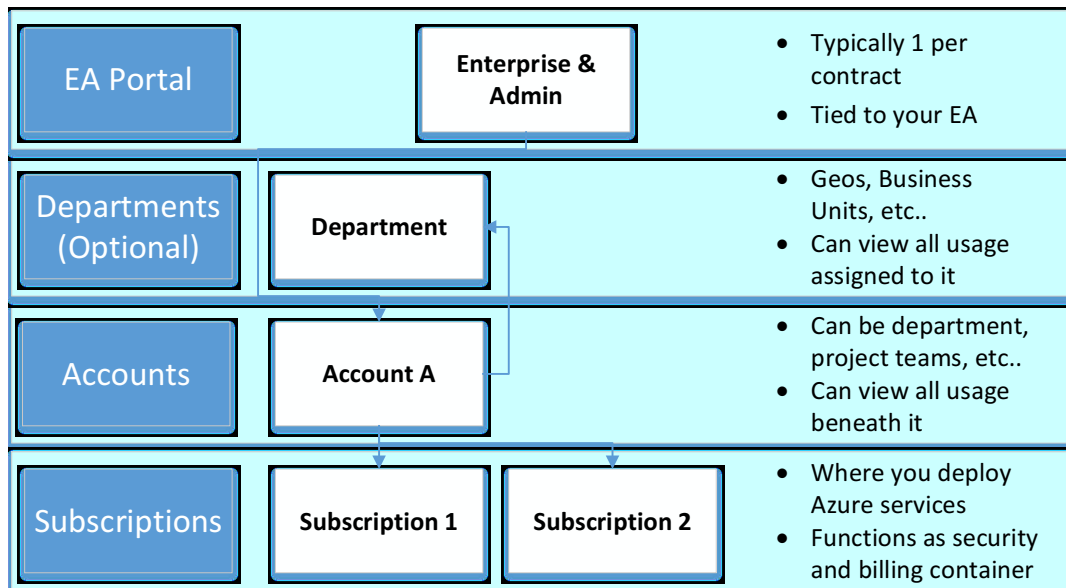


Figure 4-18: Hierarchy for organizing Azure resources from the EA contract level down to individual subscriptions

Azure subscriptions have set limits on the number of certain resource types that you can create. You can find more information at the [Azure Subscription Service Limits](#) website. The site posts the default and maximum limits. You must monitor limits, and if a higher limit is needed, you'll need to make a support request to increase it.

We recommend that you provision at least one subscription for IT cloud management, where central IT topics such as identity and Azure Active Directory, hybrid networking, or encryption key management are managed. Because of the organization (multiple teams are involved in these topics), multiple subscriptions can make sense.

Azure [DevTest Labs](#) is another tool to manage compute capacity for development and testing purposes. With DevTest Labs, you template, configure, and manage pools of virtual machines (VMs) centrally, and then give access to groups of users for specific purposes like load or environment testing. You can set usage and consumption limits for each environment. Additionally, DevTest Labs has a flexible user-rights system so that only configured users can start a pool of VMs, run their tests, or configure a new environment. DevTest Labs encourages fine-grained control of lab and test environments, which fits perfectly with the capabilities of Azure to provision and start large-scale environments on demand and then stop and deallocate them just as quickly when they are no longer needed.

Another tool for capacity management is the Azure Advisor, which can help you to identify unused resources, review performance metrics for running resources, and perform immediate resizing or deallocating of identified resources to improve efficiency. [Azure Cost Management](#) (formerly Cloudyn) is a new service with which you can monitor Azure expenditure, drive organizational accountability, and optimize Azure efficiency. Table 4-8 presents links to more resources on Azure capacity management.

Table 4-8: Capacity management

Topic	Resource
Azure billing documentation	https://docs.microsoft.com/azure/billing/
Azure service limits	https://docs.microsoft.com/azure/azure-subscription-service-limits
DevTest Labs	https://docs.microsoft.com/azure/devtest-lab/devtest-lab-overview
Azure Advisor	https://docs.microsoft.com/azure/advisor/
Azure Limits	https://docs.microsoft.com/azure/azure-subscription-service-limits
Azure Cost Management	https://azure.microsoft.com/services/cost-management/

Asset and configuration management

Assets are grouped into Azure Resource Manager groups, as depicted in Figure 4-19, which are logical containers of resources for tracking and management purposes.

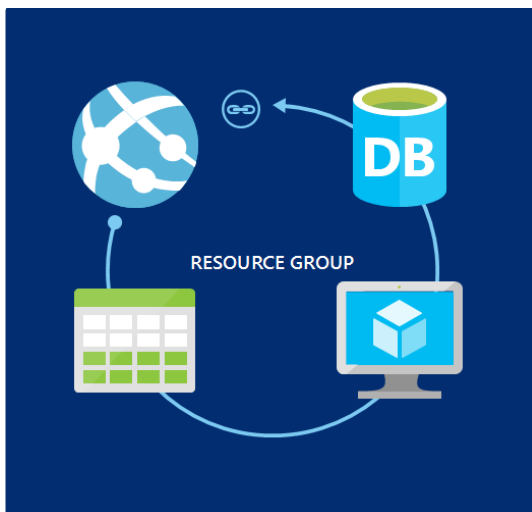


Figure 4-19: This resource group contains a web app, a data table, a VM, and a database in SQL Database

Resource groups are managed per subscription. To access them, in the Azure portal, in the pane on the left, click resource groups, as shown in Figure 4-20. You can filter resource groups by subscription and retrieve them via the Azure APIs for integration in custom management applications.

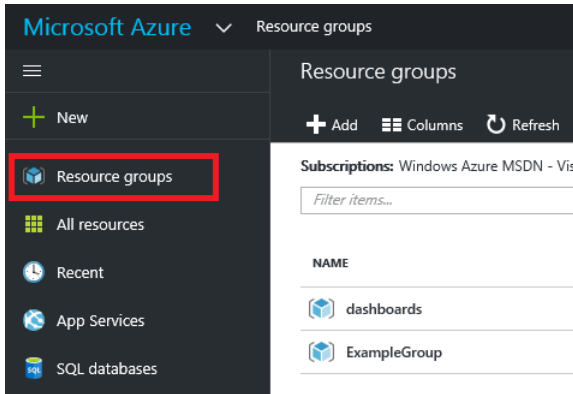


Figure 4-20: Resource groups function in the Azure portal

You can centrally manage system and device configuration for cloud, hybrid, and on-premises infrastructure in Azure by using System Center Configuration Manager, which offers strong system and infrastructure management across cloud, hybrid, and private on-premises systems.

Using Azure Automation Desired State Configuration (DSC), you can configure systems directly in the automation runbook to enforce a DSC. These jobs can store their configuration as assets in Azure Automation and be applied to systems in Azure, other clouds, or on-premises.

Chef is also a very strong tool for managing DSC; you can use a Chef Server to configure and deploy VMs with a specific configuration. For more information, follow the links in Table 4-9.

Table 4-9: Configuration management

Topic	Resource
Azure Automation DSC	https://docs.microsoft.com/azure/automation/automation-dsc-overview
System Center Configuration Manager	https://docs.microsoft.com/sccm/core/understand/configuration-manager-on-azure
Chef Automation in Azure	https://docs.microsoft.com/azure/virtual-machines/windows/chef-automation

Update and patch management

Updating and patching cloud infrastructure depends on the type of cloud resource being considered.

Azure platform

The Azure platform and its underlying infrastructure are updated via deployment rings, as illustrated in Figure 4-21, starting with inner test rings, and then gradually rolled out to datacenters worldwide.

This deployment ring roll-out strategy helps to prevent update issues from reaching users in the outer production rings because you can catch them early in the inner test and pilot rings. If an issue does occur because of Azure platform updates, any service interruptions are covered by the SLA of the relevant Azure services in a region, and incident information is published on Azure Update.

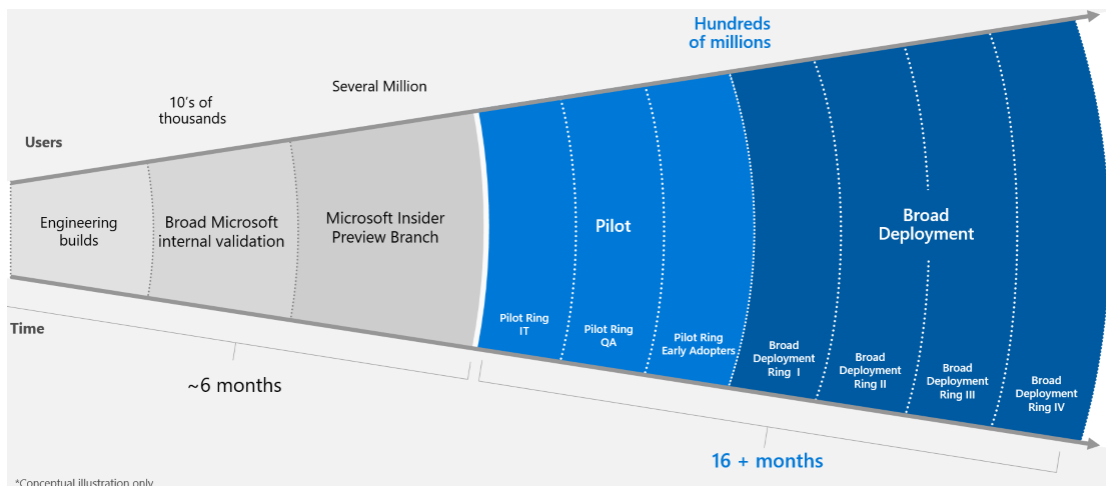


Figure 4-21: Azure deployment rings

The Azure platform itself is not versioned to the public. It is a stable and constant basis for all Azure services and maintains compatibility to the higher-level Azure services.

Software as a service

Software as a service (SaaS) products, such as Microsoft Office 365, are completely patched and updated by the software provider (including operating systems and related components). Any downtime caused by updates are covered by the SLA for the software.

Platform as a service

Azure platform as a service (PaaS) products are patched and updated by Microsoft; however, any software running on the platform must be updated and patched by the operations team responsible for the deployment.

The same principle of ring deployment is taken for Azure PaaS services. This minimizes the risk that errors or problems make it beyond the internal rings into the outer production environments. If any unforeseen unavailability of PaaS platform services should occur through a rolling update, it is covered through the PaaS platform's SLA.

PaaS platform libraries such as Java, .NET, PHP, or Python, which are included in Azure PaaS platforms such as Web Apps, are maintained in the currently available major versions, each patched to the latest minor and patched version. For this reason, it is important that you always test deployed PaaS services using the latest patch from the platform libraries to avoid problems from PaaS platform updates.

If a major version of a platform library such as Java or PHP should be deprecated from a PaaS platform, this would have major consequences for customers and would be announced in advance to give them the chance to move to a supported version.

Infrastructure as a service

Customers are responsible for patching their VMs that are deployed on Azure. A solution to help with this is Azure Update Management and Azure Automation, which utilize an agent that runs on both Windows and Linux to schedule, deliver, and install OS patches and updates.

The Update Management agent runs the update plan using the OS update mechanism, meaning either with Windows Update or a Windows Server Update Services instance for Windows Servers, or through Yum, Apt, or Zypper on Linux.

Update Management gives an overview of the patch status of all configured clients in the dashboard, as shown in Figure 4-22. You can schedule updates for a specific time period for a group of clients.

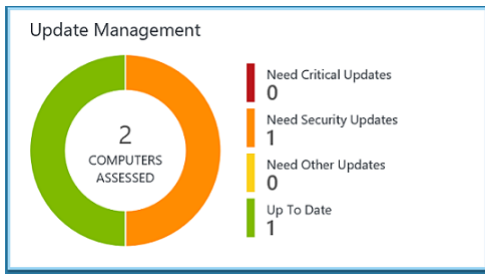


Figure 4-22: Update Management overview dashboard

Azure Log Analytics has the logs from all updates for searching, alerts, and diagnostic purposes, as shown in Figure 4-23.

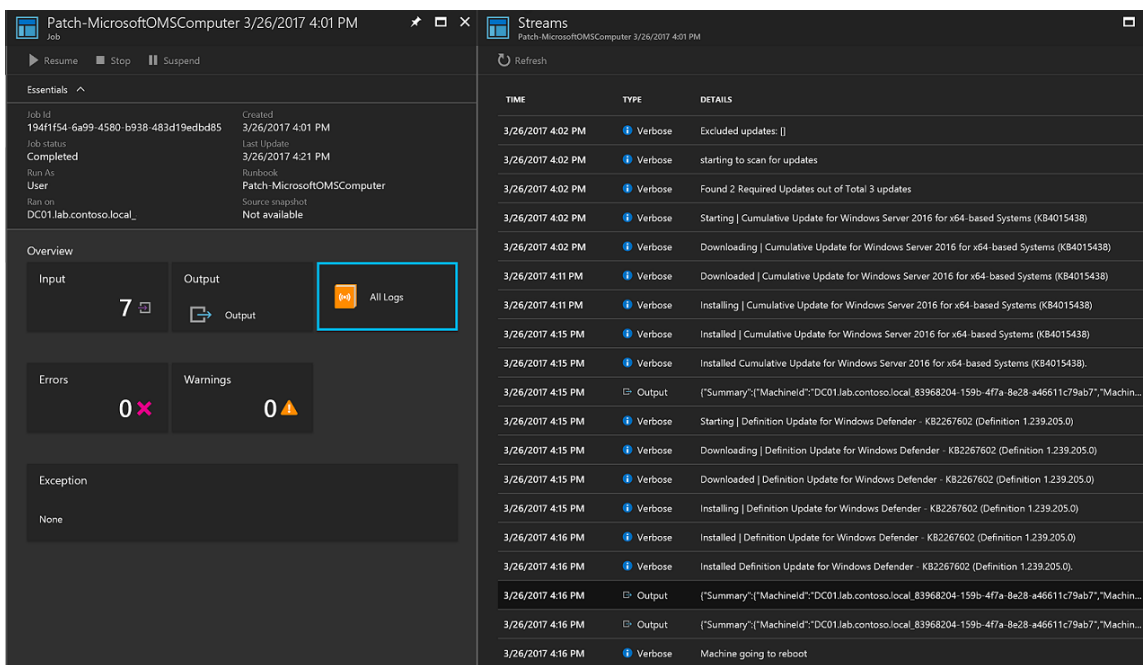


Figure 4-23: Azure Log Analytics with Update Management logs

In the end, you are responsible for patching and maintaining your IaaS VMs, but tools such as Update Management can help you to organize and optimize the patching process. Table 4-10 provides a link where you can read more about update and patch management.

Table 4-10: Update and patch management

Topic	Resource
Azure Update Management	https://docs.microsoft.com/azure/automation/automation-update-management

Service-level management

In Azure, the [SLA](#) describes Microsoft's commitments for uptime and connectivity. If the SLA for a particular service is 99.9%, it means that you should expect the service to be available 99.9% of the time.

You should define your own target SLAs for each workload in your solution. An SLA makes it possible to evaluate whether the architecture meets the business requirements. For example, if a workload requires 99.99% uptime but depends on a service with a 99.9% SLA, that service cannot be a single point of failure in the system. One remedy is to have a fallback path in case the service fails; another is to take other measures to recover from a failure in that service. Table 4-11 shows the potential cumulative downtime for various SLA levels.

Table 4-11: SLA management

SLA	Downtime per week	Downtime per month	Downtime per year
99%	1.68 hours	7.2 hours	3.65 days
99.9%	10.1 minutes	43.2 minutes	8.76 hours
99.95%	5 minutes	21.6 minutes	4.38 hours
99.99%	1.01 minutes	4.32 minutes	52.56 minutes
99.999%	6 seconds	25.9 seconds	5.26 minutes

Of course, higher availability is better, everything else being equal. But as you strive for more 9's, the cost and complexity to achieve that level of availability grows. An uptime of 99.99% translates to about five minutes of total downtime per month. Is it worth the additional complexity and cost to reach five 9's? The answer depends on the business requirements. Here are some other considerations when defining an SLA:

- To achieve four 9's (99.99%), you can't rely on manual intervention to recover from failures. The application must be self-diagnosing and self-healing.
- Beyond four 9's, it is challenging to detect outages quickly enough to meet the SLA.
- Think about the time window against which your SLA is measured. The smaller the window, the tighter the tolerances. It probably doesn't make sense to define your SLA in terms of hourly or daily uptime

Composite SLAs

Consider an App Service web app that writes to SQL Database. As of this writing, these Azure services have the following SLAs:

- App Service Web Apps = 99.95%
- SQL Database = 99.99%

If either service fails, the entire application fails. In general, the probability of each service failing is independent, so the composite SLA for this application is $99.95\% \times 99.99\% = 99.94\%$. That's lower than the individual SLAs, which isn't surprising, because an application that relies on multiple services has more potential failure points. On the other hand, you can improve the composite SLA by creating independent fallback paths. For example, if SQL Database is unavailable, put transactions into a queue to be processed later, as illustrated in Figure 4-24.

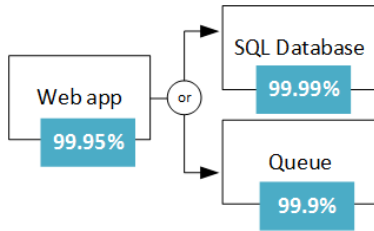


Figure 4-24: Composite SLA

With this design, the application is still available even if it can't connect to the database. However, it fails if the database and the queue both fail at the same time. The expected percentage of time for a simultaneous failure is 0.0001×0.001 , so the composite SLA for this combined path is as follows:

$$\text{Database OR queue} = 1.0 - (0.0001 \times 0.001) = 99.99999\%$$

Here's the total composite SLA:

$$\text{Web app AND (database OR queue)} = 99.95\% \times 99.99999\% = \sim 99.95\%$$

But there are trade-offs to this approach: The application logic is more complex, you are paying for the queue, and there can be data consistency issues to consider.

SLA for multiregion deployments

Another high-availability technique is to deploy the application in more than one region and use Azure Traffic Manager to failover if the application fails in one region. For a two-region deployment, the composite SLA is calculated as follows:

- Let N be the composite SLA for the application deployed in one region. The expected chance that the application will fail in both regions at the same time is $(1 - N) \times (1 - N)$; therefore:

$$\text{Combined SLA for both regions} = 1 - (1 - N)(1 - N) = N + (1 - N)N$$

- Finally, you must factor in the [SLA for Traffic Manager](#). As of this writing, the SLA for Traffic Manager SLA is 99.99%.

$$\text{Composite SLA} = 99.99\% \times (\text{combined SLA for both regions})$$

Also, failing-over is not instantaneous and can result in some downtime during a failover. See Traffic Manager endpoint monitoring and failover.

The calculated SLA number is a useful baseline, but it doesn't tell the entire story about availability. Often, an application can degrade gracefully when a noncritical path fails.

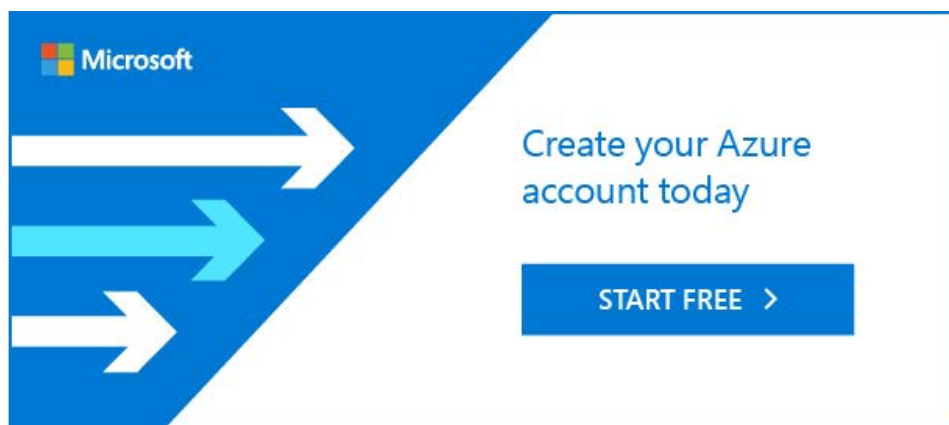
Conclusion

The potential of cloud computing for all organizations is immense, but it also requires changes in governance, architecture, operations and service management. We have tried to show in this document that not everything needs to change; however, adjustments and new ideas in key areas can unlock a lot of the potential and make cloud transformation in any IT organization possible.

Here are some key points from each of the chapters:

- Governance and a “cloud champions” team to drive the transformation can help bring the attention needed in an organization. This is best accomplished with executive sponsorship.
- Architecture for cloud-first applications incorporates security, identity, and modern application design for robust apps and data that are available for users anywhere from any device.
- Application development together with operations—DevOps—should be the default for new projects. The DevOps model fits with cloud technologies to effectively develop and operate cloud-first apps that users value and want to use.
- Integrating, planning, and managing cloud resources directly in the service management (instead of in shadow IT) helps both the business and IT organization profit from the flexibility of cloud infrastructure.

For certain, the change will not stop and the drive for ever more efficient and advanced cloud infrastructure will continue to make it possible for teams to deliver increasingly advanced products at a faster pace. The teams and organizations that profit the most will be those that fully utilize and integrate cloud platforms, thereby freeing them up to spend most of their time delivering customer value. Taking advantage of the cloud and the effective implementation and integration of cloud technologies is a key part of the transformation to win the battle for users’ hearts, minds, and, most important, their screen time.



About the authors



Joachim Hafner is a cloud solution architect at Microsoft Germany, helping clients to integrate the Azure platform into their enterprise architecture and providing guidance for designing modern cloud applications based on Azure technologies. Before he joined Microsoft, he worked as a senior enterprise architect for one of the largest IT service providers with a strong focus on hybrid cloud strategies.



Simon Schwingel is a cloud solution architect at Microsoft Germany, where he assists customers to move to the cloud. He provides guidance on how to unleash the potential of public cloud-based architectures and technologies. Simon started his career 19 years ago as a web developer. Since then, he performed *inter alia* as consultant for Enterprise Content Management solutions and as lead architect for cloud-based collaboration services for one of the largest IT service providers, with a focus on private cloud infrastructure.



Tyler Ayers is a cloud solution architect at Microsoft Germany, working with customers to define and integrate their cloud strategy with Azure in the financial services and media industries. Previously, he worked as lead architect for enterprise software products in the retail and banking industries and was an early proponent for integrating cloud and hybrid technologies into every IT organization's toolbox.



Rolf McLaughlin (MASUCH) is a cloud solution architect at Microsoft Germany with a focus on Azure Governance, Infrastructure, and Security. Rolf started his career as a Microsoft Certified Trainer for Windows in 1996. While maintaining his status as a trainer, he worked for various Microsoft partners in several roles, mainly in the consulting area around large Active Directory consolidations and implementing large-scale messaging systems based on Microsoft Exchange. He also was honored with the status of Most Valuable Professional (MVP) for his efforts in the PowerShell community.



Azure Sales Desk

For questions about getting started, or for tips and tricks on how to use and optimize Microsoft Azure services, please contact the Azure Sales Desk. Please note that Azure support questions should be directed to technical support.

[Email us](#)

[Get technical support](#)

PUBLISHED BY
Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Copyright © 2018 by Microsoft Corporation