AZURE WELL-ARCHITECTED FRAMEWORK

Azure Advisor

Advisor is a personalized cloud consultant that helps you follow best practices to optimize your Azure deployments. It analyzes your resource configuration and usage telemetry and then recommends solutions that can help you improve the cost effectiveness, performance, reliability (formerly called high availability) and security of your Azure resources.

The Advisor dashboard displays personalized recommendations for all your subscriptions. You can apply filters to display recommendations for specific subscriptions and resource types.

WITH ADVISOR, YOU CAN:

- Get proactive, actionable and personalized best practices recommendations.
- Improve the performance, security and reliability of your resources, as you identify opportunities to reduce your overall Azure spend.
- Get recommendations with proposed actions inline.

The recommendations are divided into five categories based on the Well-Architected Framework pillars:

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

COST OPTIMIZATION

When you are designing a cloud solution, focus on generating incremental value early. Apply the principles of Build-Measure-Learn, to accelerate your time to market while avoiding capital-intensive solutions. Use the pay-as-you-go strategy for your architecture, and invest in scaling out, rather than delivering a large investment first version. Consider opportunity costs in your architecture, and the balance between first-mover advantage versus "fast follow". Use the cost calculators to estimate the initial cost and operational costs. Finally, establish policies, budgets and controls that set cost limits for your solution.

OPERATIONAL EXCELLENCE

This pillar covers the operations and processes that keep an application running in production. Deployments must be reliable and predictable. They should be automated to reduce the chance of human error. They should be a fast and routine process, so they don't slow down the release of new features or bug fixes. Equally important, you must be able to quickly roll back or roll forward if an update has problems.

Monitoring and diagnostics are crucial. Cloud applications run in a remote data centre where you do not have full control of the infrastructure or, in some cases, the operating system. In a large application, it's not practical to log into virtual machines (VMs) to troubleshoot an issue or sift through log files. With PaaS services, there may not even be a dedicated VM to log into. Monitoring and diagnostics give insight into the system, so that you know when and where failures occur. All systems must be observable. Use a common and consistent logging schema that lets you correlate events across systems.

PERFORMANCE EFFICIENCY

Performance efficiency is the ability of your workload to scale to meet the demands placed on it by users in an efficient manner. The main ways to achieve this are by using scaling appropriately and implementing PaaS offerings that have scaling built in.

There are two main ways that an application can scale. Vertical scaling (scaling up) means increasing the capacity of a resource, for example, by using a larger VM Size. Horizontal scaling (scaling out) is adding new instances of a resource, such as VMs or database replicas.

Horizontal scaling has significant advantages over vertical scaling:

- True cloud scale. Applications can be designed to run on hundreds or even thousands of nodes, reaching scales that are not possible on a single node.
- Horizontal scale is elastic. You can add more instances if the load increases or remove them during quieter periods.
- Scaling out can be triggered automatically, either on a schedule or in response to changes in load.
- Scaling out may be cheaper than scaling up. Running several small VMs can cost less than a single large VM.
- Horizontal scaling can also improve resiliency, by adding redundancy. If an instance goes down, the application keeps running.

An advantage of vertical scaling is that you can do it without making any changes to the application. But at some point, you'll hit a limit where you can't scale any up anymore. At that point, any further scaling must be horizontal.

RELIABILITY

A reliable workload is one that is both resilient and available. Resiliency is the ability of the system to recover from failures and continue to function. The goal of resiliency is to return the application to a fully functioning state after a failure occurs. Availability is whether your users can access your workload when they need to.

In traditional application development, there has been a focus on increasing the mean time between failures (MTBF). An effort was spent trying to prevent the system from failing. In cloud computing, a different mindset is required, due to several factors:

- Distributed systems are complex and a failure at one point can potentially cascade throughout the system.
- Costs for cloud environments are kept low through the use of commodity hardware, so occasional hardware failures must be expected.
- Applications often depend on external services, which may become temporarily unavailable or throttle high-volume users.
- Today's users expect an application to be available 24/7 without ever going offline.

All of these factors mean that cloud applications must be designed to expect occasional failures and recover from them. Azure has many resiliency features already built into the platform. For example:

- Azure Storage, SQL Database, and Cosmos DB all provide built-in data replication, both within a region and across regions.
- Azure-managed disks are automatically placed in different storage scale units to limit the effects of hardware failures.
- VMs in an availability set are spread across several fault domains. A fault domain is a group of VMs that share a common power source and network switch. Spreading VMs across fault domains limits the impact of physical hardware failures, network outages, or power interruptions.

That said, you still need to build resiliency into your application. Resiliency strategies can be applied at all levels of the architecture. Some mitigations are more tactical in nature — for example, retrying a remote call after a transient network failure. Other mitigations are more strategic, such as failing over the entire application to a secondary region. Tactical mitigations can make a big difference. While it's rare for an entire region to experience a disruption, transient problems such as network congestion are more common — so target these first. Having the right monitoring and diagnostics is also important, both to detect failures when they happen and to find the root causes.

When designing an application to be resilient, you must understand your availability requirements. How much downtime is acceptable? This is partly a function of cost. How much will potential downtime cost your business? How much should you invest in making the application highly available?

SECURITY

Think about security throughout the entire lifecycle of an application, from design and implementation to deployment and operations. The Azure platform provides protection against a variety of threats, such as network intrusion and DDoS attacks. But you still need to build security into your application and into your DevOps processes.

Here are some broad security areas to consider:

- **IDENTITY MANAGEMENT**

  Consider using Azure Active Directory (Azure AD) to authenticate and authorize users. Azure AD is a fully managed identity and access management service. You can use it to create domains that exist purely on Azure or integrate with your on-premises Active Directory identities. Azure AD also integrates with Office365, Dynamics CRM Online and many third-party SaaS applications. For consumer-facing applications, Azure Active Directory B2C lets users authenticate with their existing social accounts (such as Facebook, Google, or LinkedIn), or create a new user account that is managed by Azure AD.

  If you want to integrate an on-premises Active Directory environment with an Azure network, several approaches are possible, depending on your requirements.

- **PROTECTING YOUR INFRASTRUCTURE**

  Control access to the Azure resources that you deploy. Every Azure subscription has a trust relationship with an Azure AD tenant. Use Azure role-based access control (Azure RBAC) to grant users within your organization the correct permissions to Azure resources. Grant access by assigning Azure roles to users or groups at a certain scope. The scope can be a subscription, a resource group, or a single resource. Audit all changes to infrastructure.

- **APPLICATION SECURITY**

  In general, the security best practices for application development still apply in the cloud. These include things like using SSL everywhere, protecting against CSRF and XSS attacks, preventing SQL injection attacks, and so on.

  Cloud applications often use managed services that have access keys. Never check these into source control. Consider storing application secrets in Azure Key Vault.

- **DATA SOVEREIGNTY AND ENCRYPTION**

  Make sure that your data remains in the correct geopolitical zone when using Azure data services. Azure's geo-replicated storage uses the concept of a paired region in the same geopolitical region.

- Use Key Vault to safeguard cryptographic keys and secrets. By using Key Vault, you can encrypt keys and secrets by using keys that are protected by hardware security modules (HSMs). Many Azure storage and DB services support data encryption at rest, including Azure Storage, Azure SQL Database, Azure Synapse Analytics and Cosmos DB.

For more information, contact your CDW account team at 800.972.3922 or visit CDW.ca/azure.

CDW® PEOPLE WHO GET IT™