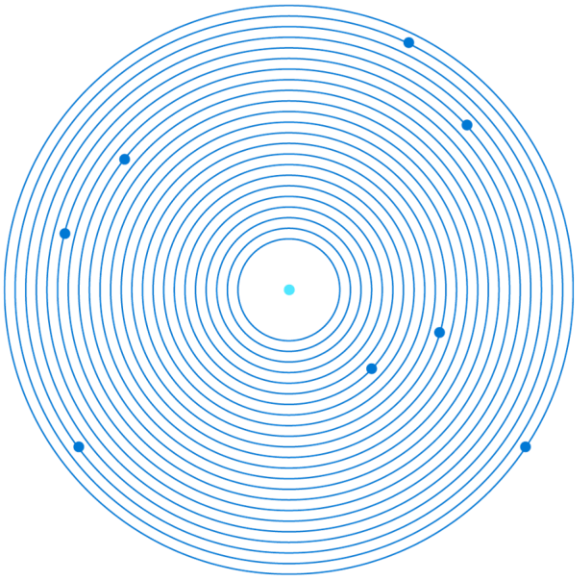
Microsoft Azure

Designing an Azure Data Solution [DP-201]



1

Agenda

- About this course
- Audience
- Course agenda
- Prerequisites

2

About this course

In this course, the students will design various data platform technologies into solutions that are in line with business and technical requirements. This can include on-premises, cloud, and hybrid data scenarios which incorporate relational, No-SQL or Data Warehouse data. They will also learn how to design process architectures using a range of technologies for both streaming and batch data

The students will also explore how to design data security including data access, data policies and standards. They will also design Azure data solutions which includes the optimization, availability and disaster recovery of big data, batch processing and streaming data solutions

3

Course agenda

Module 01 Architecture considerations	Module 2 Azure batch processing reference architectures
Lesson 01 – Describe the core principles for creating architectures	Lesson 01 – Describe Lambda architectures from a Batch Mode Perspective
Lesson 02 – Design with Security in mind	Lesson 02 – Design an Enterprise BI solution in Azure
Lesson 03 – Consider performance and scalability	Lesson 03 – Automate enterprise BI solutions in Azure
Lesson 04 – Design for availability and recoverability	Lesson 04 – Architect an Enterprise-grade conversational bot in Azure
Lesson 05 – Design for efficiency and operations	
Lesson 06 – Understand the course Case Study	

4

Course agenda (*continued #1*)

Module 03
Azure real-time reference architectures

- Lesson 01 – Lambda architectures for a real-time mode perspective
- Lesson 02 – Architect a stream processing pipeline with Azure Stream Analytics
- Lesson 03 – Design a stream processing pipeline with Azure Databricks
- Lesson 04 – Create an Azure IoT reference architecture

Module 04
Security design considerations

- Lesson 01 – Defense in depth security approach
- Lesson 02 – Identity protection
- Lesson 03 – Infrastructure protection
- Lesson 04 – Encryption usage
- Lesson 05 – Network level protection
- Lesson 06 – Application security

5

Course agenda (*continued #2*)

Module 05
Designing for scale and resiliency

- Lesson 01 – Adjust workload capacity by scaling
- Lesson 02 – Optimize network performance
- Lesson 03 – Design for optimized storage and database performance
- Lesson 04 – Identifying performance bottlenecks
- Lesson 05 – Design a highly available solution
- Lesson 06 – Incorporate disaster recovery into architectures
- Lesson 07 – Design backup and restore strategies

Module 06
Design for efficiency and operations

- Lesson 01 – Maximize the efficiency of your cloud environment
- Lesson 02 – Use monitoring and analytics to gain operational insights
- Lesson 03 – Use automation to reduce effort and error

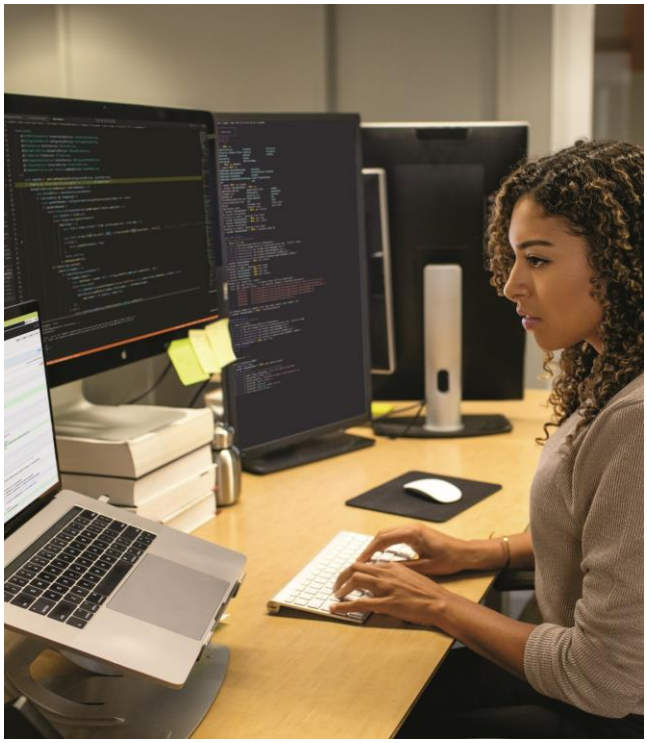
6

Audience

Primary audience:
The audience for this course is data professionals, data architects, and business intelligence professionals who want to learn about the data platform technologies that exist on Microsoft Azure

Secondary audience:
The secondary audience for this course is individuals who develop applications that deliver content from the data platform technologies that exist on Microsoft Azure

7



Prerequisites

In addition to their professional experience, students who take this training should have technical knowledge equivalent to the following courses:

Azure fundamentals

DP200: Implementing an Azure Data Solution

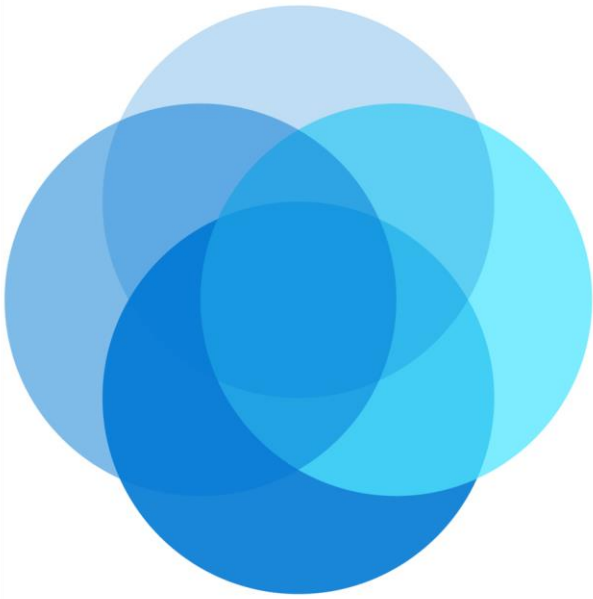
8

Microsoft Azure

9

Microsoft Azure

Module 01: Azure architecture considerations



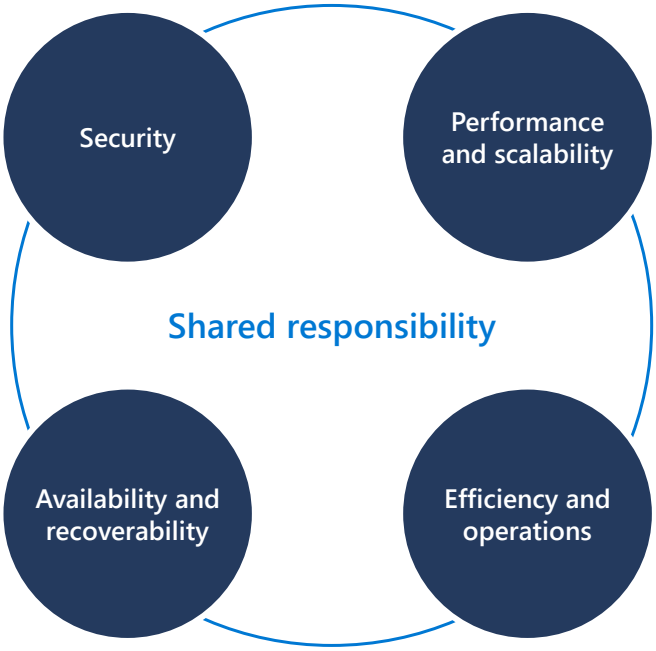
10

Lesson objectives

- Describe the pillars of a great Azure architecture
- Design with security in mind
- Consider performance and scalability
- Design for availability and recoverability
- Design for efficiency and operations
- Understand the course case study

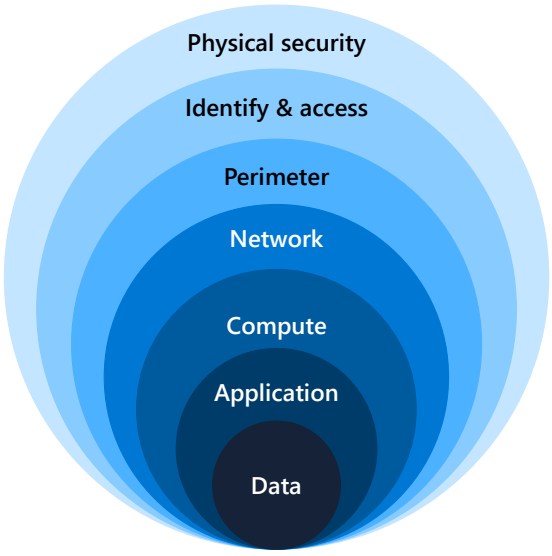
11

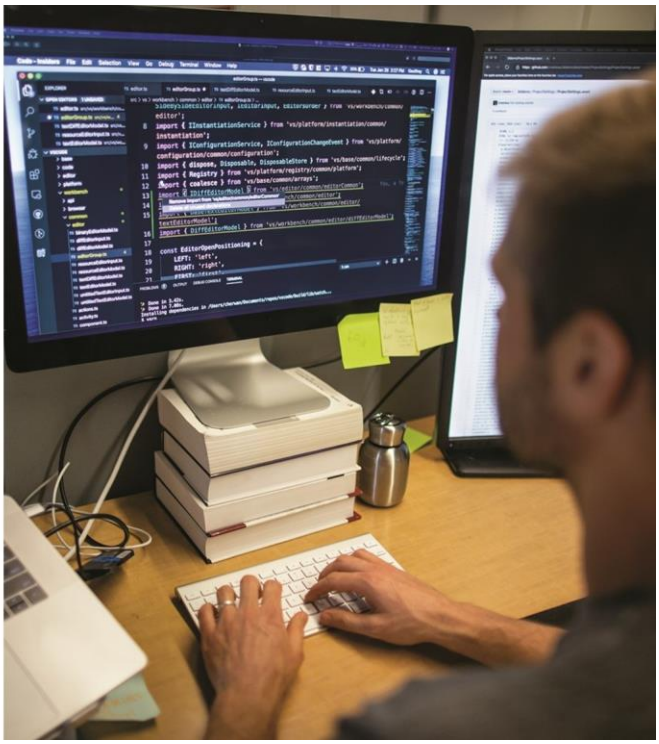
Pillars of a great Azure architecture



12

Design for security





13

Design for performance and scalability

Scaling:
Compute resources can be scaled in two different directions:
- Scaling **UP** is the action of adding more resources to a single instance
- Scaling **out** is the addition of instances

Performance:
When optimizing for performance, you'll look at network and storage to ensure performance is acceptable. Both can impact the response time of your application and databases

Patterns and practices

Partitioning:
In many large-scale solutions, data is divided into separate partitions that can be managed and accessed separately

Scaling:
Is the process of allocating scale units to match performance requirements. This can be done either automatically or manually

Caching:
Caching is a mechanism to store frequently used data or assets (web pages, images) for faster retrieval

14

Design for availability and recoverability

Design for availability:

Designing for *availability* focuses on maintaining uptime through small-scale incidents and temporary conditions like partial network outages

Design for recoverability:

Designing for *recoverability* focuses on recovery from data loss and from larger scale disasters

Recovery point objective:
The maximum duration of acceptable data loss

Recovery time objective:
The maximum duration of acceptable downtime

15

Design for efficiency and operations

Importance of efficiency and operations:

Efficiency is focused on identifying and eliminating waste within your environment. The cloud is a pay-as-you-go service and waste typically comes from provisioning more capacity than demand requires. There are operational costs that go along with this as well. These operational costs show up as wasted time and increased error

Efficiency best practices:

Look at cost optimization steps like sizing data services or virtual machines properly and deallocating compute that aren't in use. Now that you are paying for what you use, you want to be sure that you aren't wasting any of these resources

Operational best practices:

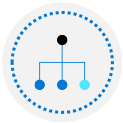
Automate as much as possible. The human element is costly, injecting time and error into operational activities. You can use automation to build, deploy, and administer resources. By automating common activities, you can eliminate the delay in waiting for a human to intervene

16

Course case study: AdventureWorks Cycles

Read the case study

In this section of the course, the instructor will either:



Allocate you 10 minutes to read through the case study

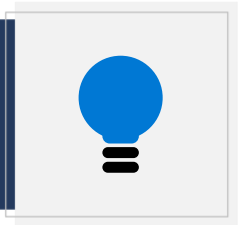
OR

Spend 10 minutes walking through the case study with you as a group

Note:
This case study will be used in labs across the entire course. Each lab will drill down more into the detail of what is required as you perform each lab

17

Lab: Architecture considerations



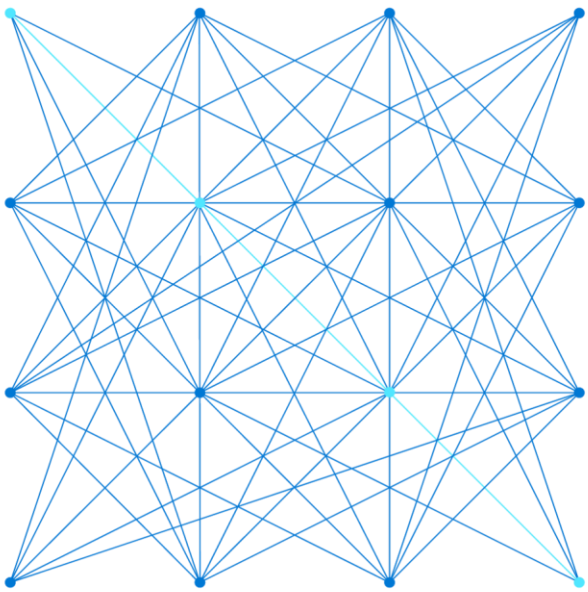
18

Microsoft Azure

19

Microsoft Azure

Module 02: Azure batch processing reference architectures



20

Lesson objectives

- Lambda architectures from a batch mode perspective
- Design an enterprise business intelligence solution
- Automate an enterprise business intelligence solution
- Architect an enterprise conversational bot

21

Lambda architectures from a batch mode perspective

Batch Layer:

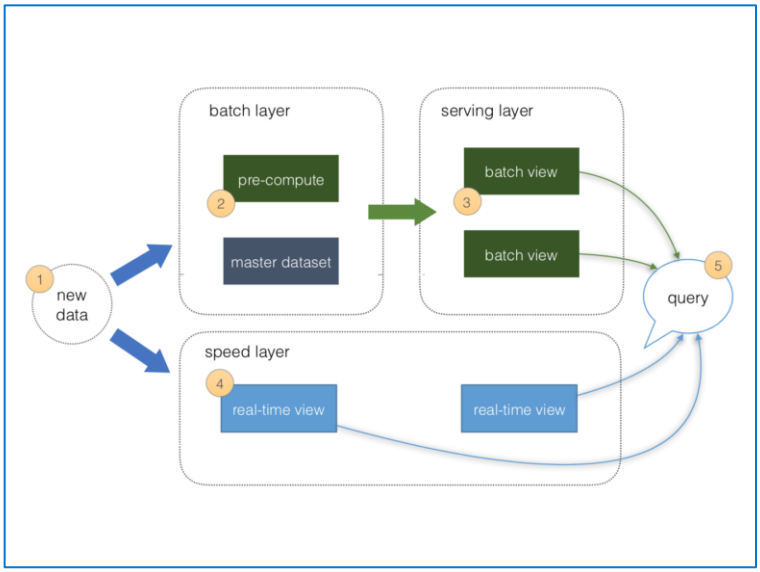
The Batch layer is used to pre-compute results on large volumes of data using a distributed data system:

1. New data ingested from sources
2. Data is precomputed

Serving Layer:

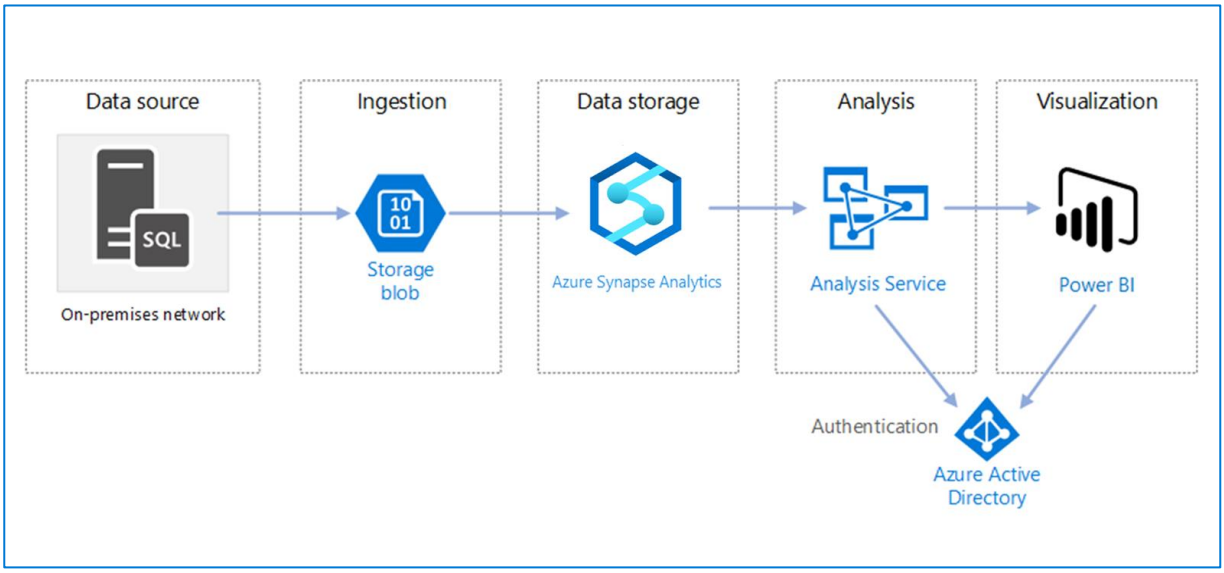
The serving layer acts as the storage output of either the Batch or Speed layer that is used by client applications to access the results of the data-sets:

3. Batch views can be used as precomputed views



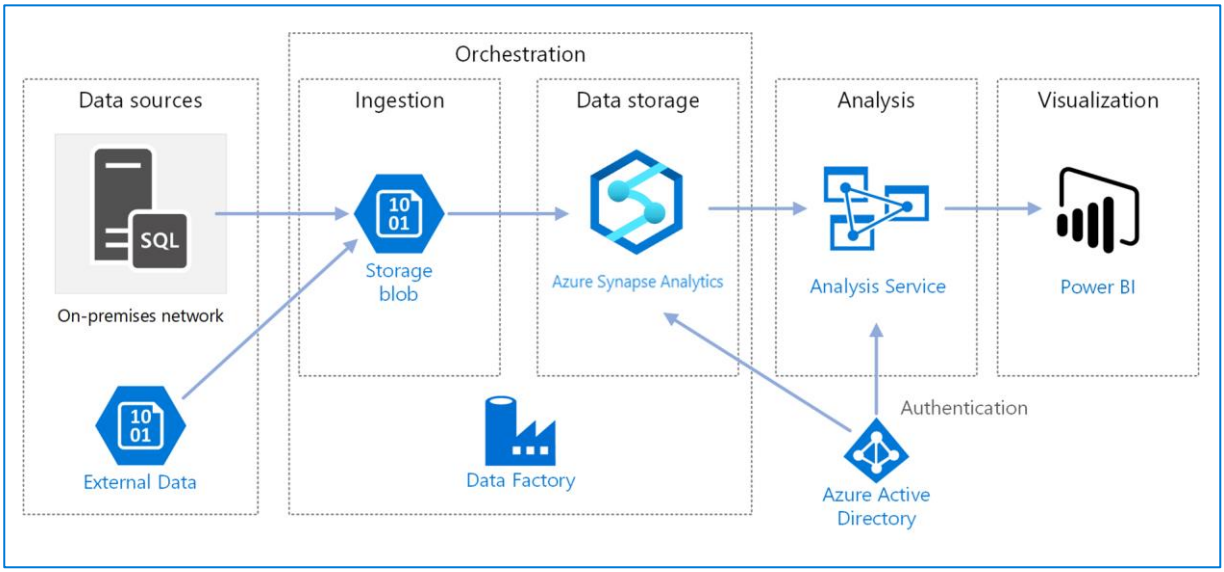
22

Design an enterprise business intelligence architecture



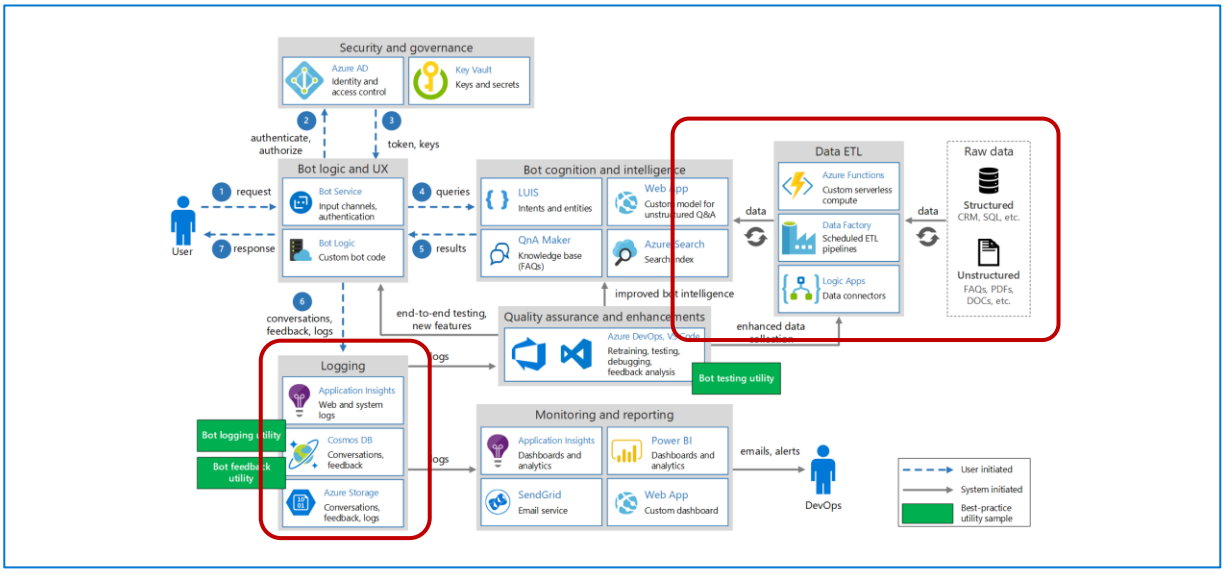
23

Automate an enterprise business intelligence architecture



24

Architect an enterprise conversational bot



25

Lab: Azure batch processing reference architectures

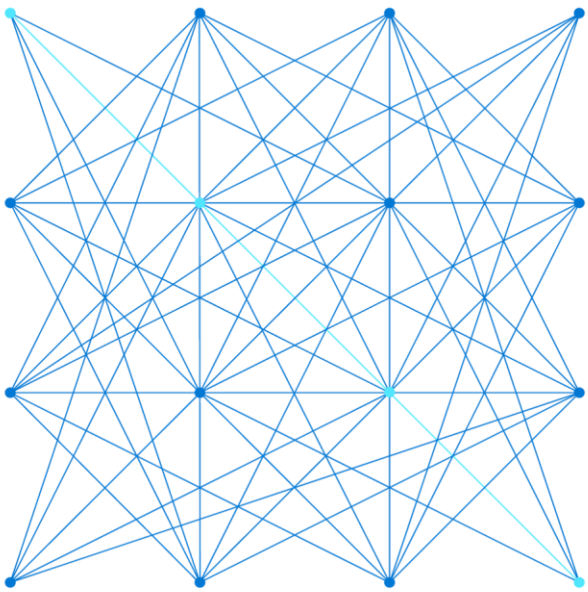
26

Microsoft Azure

27

Microsoft Azure

Module 03: Azure real-time processing reference architectures



28

Lesson objectives

- Lambda architectures for a real-time mode perspective
- Architect a stream processing pipeline with Azure Stream Analytics
- Design a stream processing pipeline with Azure Databricks
- Create an Azure IoT reference architecture

29

Lambda architectures from a real time mode perspective

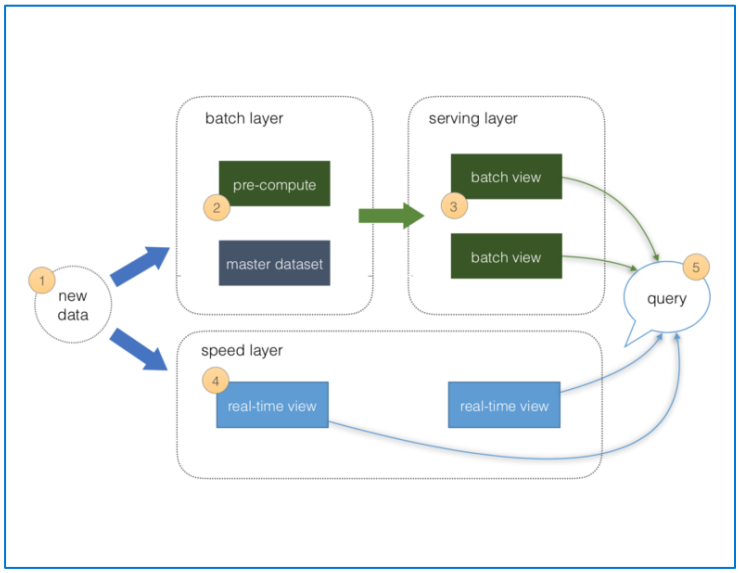
Speed Layer:

The Speed layer processes data streams in real or near real time. This works well when the aim is to minimize the latency of the data ingestion to analysis:

1. New data ingested from sources
4. Real time views of the data created

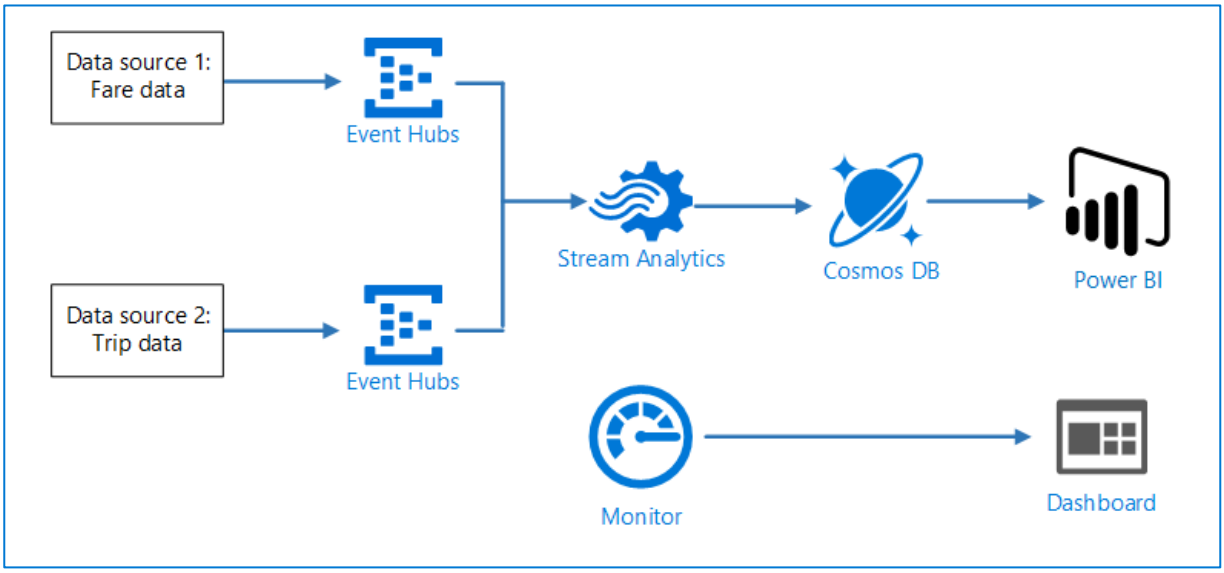
Serving Layer:

The serving layer is optional in the real-time architecture and acts as the storage output of either the Batch or Speed layer that is used by client applications to access the results of the data-sets



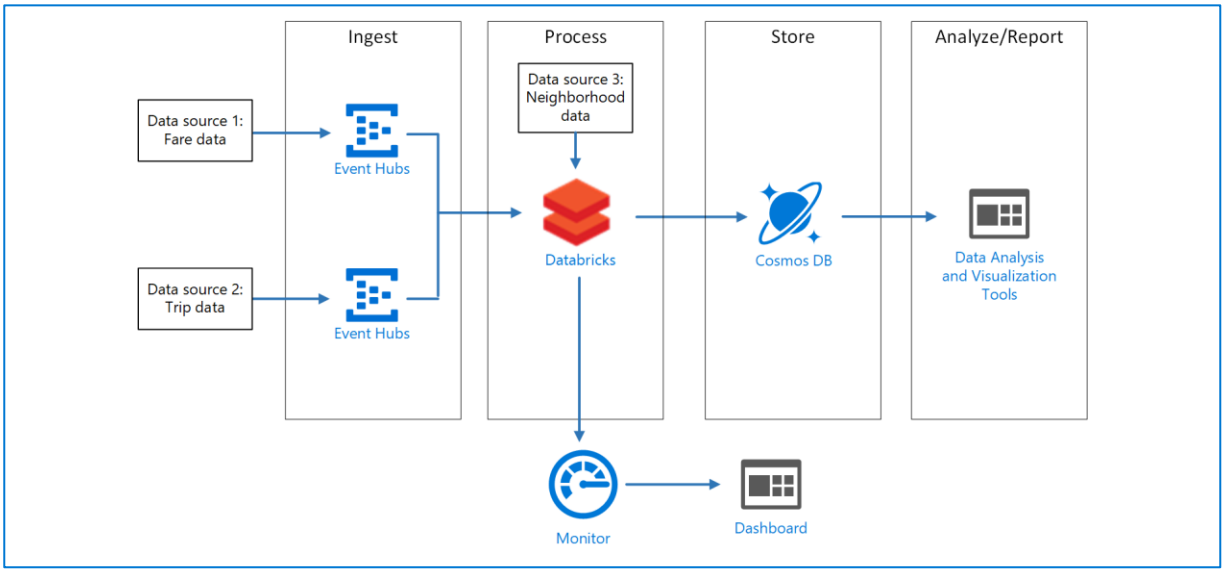
30

Architect a stream processing pipeline with Azure Stream Analytics



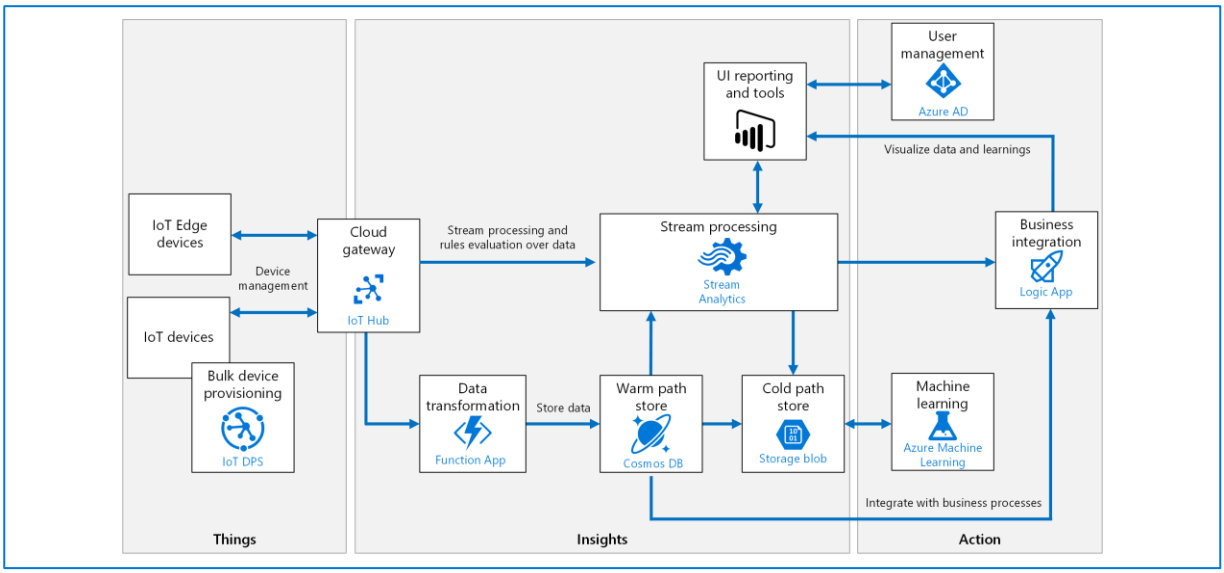
31

Design a stream processing pipeline with Azure Databricks



32

Create an Azure IoT reference architecture



33

Lab: Azure real-time processing reference architectures

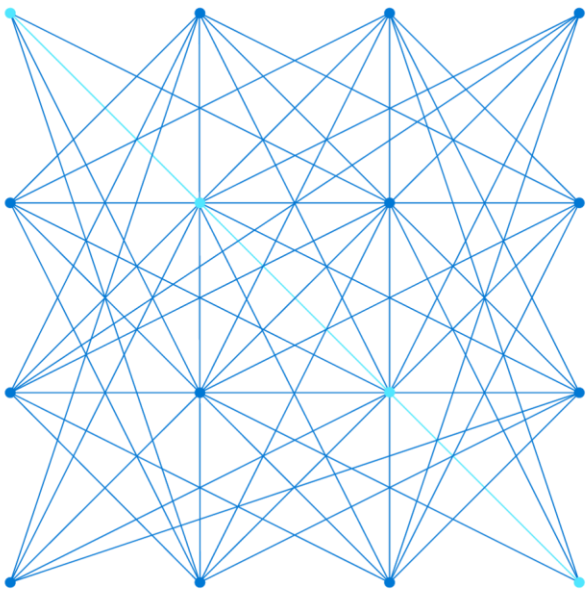
34

Microsoft Azure

35

Microsoft Azure

Module 04: Security design considerations



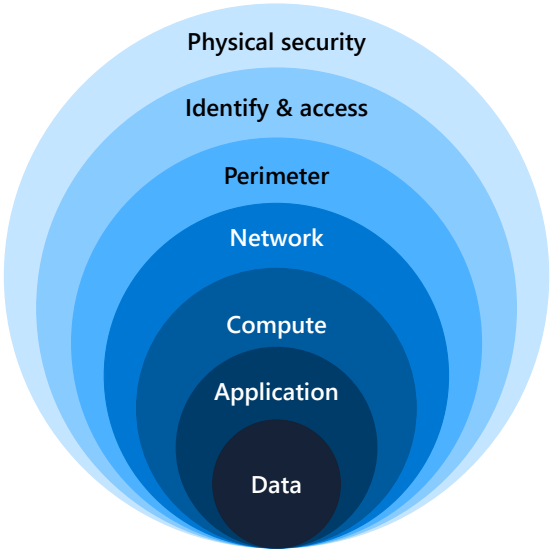
36

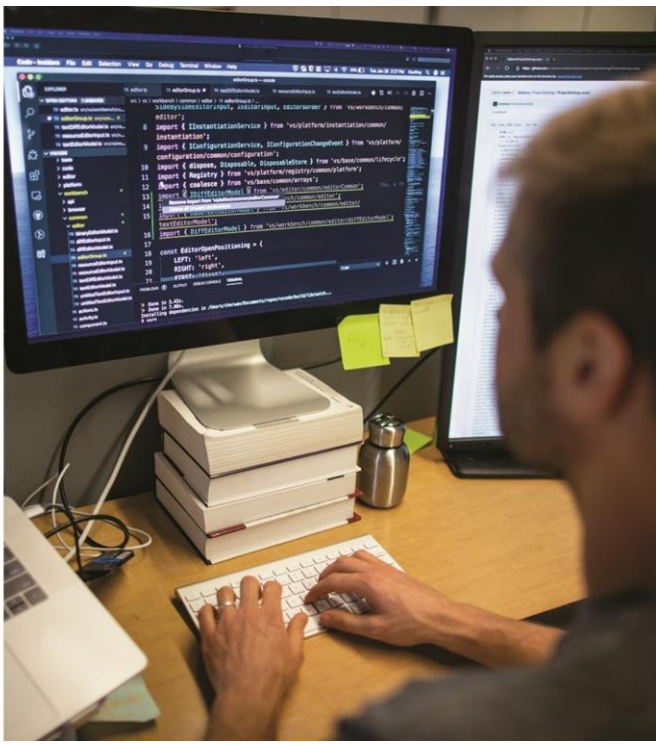
Lesson objectives

- Take a defense in depth approach to securing your architecture
- How to protect your identities
- Technologies to protect your Azure infrastructure
- Use encryption to secure your data
- Protect your architecture at the network level
- Leverage application security best practices

37

Defence in depth approach





38

Identity management

Identifying users that access your resources is an important part of security design

Identity as a security layer:

Customers and employees alike expect to be able to access services from anywhere at any time, which has driven the development of identity protocols that can work at internet scale across many disparate devices and operating systems

Single sign-on:

With single sign-on, users only need to remember one ID and one password. Access across database systems or applications is granted to a single identity tied to a user

SSO with Azure Active Directory:

Azure Active Directory (AD) is a cloud-based identity service. It has built-in support for synchronizing with your existing on-premises Active Directory or can be used stand-alone. This means that all your applications, whether on-premises, in the cloud (including Office 365), or even mobile can share the same credentials

39

Infrastructure protection

Role based access control:

Roles are defined as collections of access permissions. Security principals are mapped to roles directly or through group membership

Roles and management groups:

Roles are sets of permissions that users can be granted to. Management groups add the ability to group subscriptions together and apply policy at an even higher level

Privileged identity management:

Azure AD Privileged Identity Management (PIM) is an additional paid-for offering that provides oversight of role assignments, self-service, and just-in-time role activation

Providing identities to services:

An Azure service can be assigned an identity to ease the management of service access to other Azure resources

Service principals:

A Service Principal is literally named. It is an identity that is used by a service or application. Like other identities, it can be assigned roles

Managed identities:

When you create a managed identity for a service, you create an account on the Azure AD tenant. Azure infrastructure will automatically take care of authentication

40

Encryption

Encryption at rest:
Data at rest is the data that has been stored on a physical medium. This could be data stored on the disk of a server, data stored in a database, or data stored in a storage account

Encryption in transit:
Data in transit is the data actively moving from one location to another, such as across the internet or through a private network. Secure transfer can be handled by several different layers

Encryption on Azure

Raw encryption:
Enables the encryption of:
- Azure Storage
- V.M. Disks
- Disk Encryption

Database encryption:
Enables the encryption of databases using:
- Transparent Data Encryption

Encrypting secrets:
Azure Key Vault is a centralized cloud service for storing your application secrets

41

Network security

Network security is protecting the communication of resources within and outside of your network. The goal is to limit exposure at the network layer across your services and systems

Internet protection:
A great first place to start is to assess the resources that are internet-facing, and only allow inbound and outbound communication where necessary. Identify all resources that are allowing inbound network traffic of any type, and ensure they are necessary and restricted to only the ports/protocols required

Virtual network security:
To isolate Azure services to only allow communication from virtual networks, use VNet service endpoints. With service endpoints, Azure service resources can be secured to your virtual network

Network integration:
VPN connections are a common way of establishing secure communication channels between networks, and this is no different when working with virtual networking on Azure. Connection between Azure VNets and an on-premises VPN device is a great way to provide secure communication

42

Application security

43

Lab: Azure Data Platform security design considerations

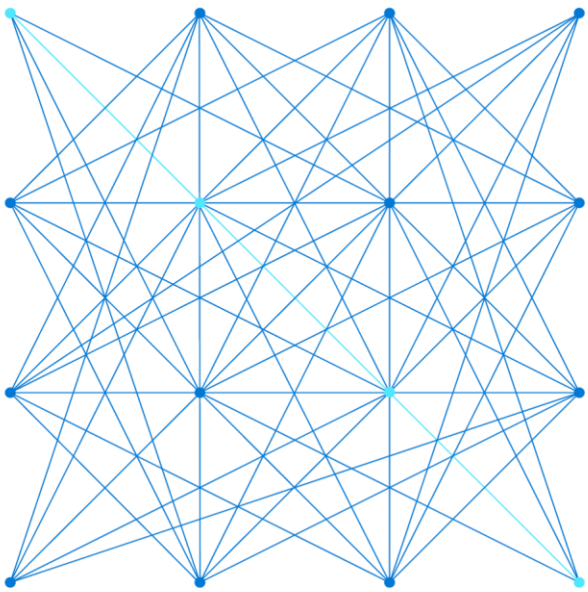
44

Microsoft Azure

45

Microsoft Azure

Module 05: Designing for scale and resiliency



46

Lesson objectives

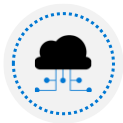
Scaling systems

High availability

Optimizing network performance

Disaster recovery

Optimizing storage performance



Backup and restore

Identifying performance bottlenecks

47

Scaling systems

Scaling up or down:

Scaling up or down is the process where we increase or decrease the capacity of a given instance. It adjusts the amount of resources a single instance has available. You can use autoscaling so that this can be managed automatically

Scaling in or out:

Scaling out is the process of adding more instances to support the load of your solution. Scaling in is the process of removing instances that are no longer needed

Serverless computing

Containers:

A container is a method running applications in a virtualized environment. The virtualization is done at the OS level, making it possible to run multiple identical application instances within the same OS

Azure Kubernetes Service (AKS):

Azure Kubernetes Service allows you to set up virtual machines to act as your nodes. Azure hosts the Kubernetes management plane and only bills for the running worker nodes that host your containers

Azure Container Instance (ACI):

Azure Container Instances is a serverless approach that lets you create and execute containers on demand. You're charged only for the execution time per second

48

Optimizing network performance

Network performance can have a dramatic impact on a user's experience. In complex architectures with many different services, minimizing the latency at each hop can have a huge impact on the overall performance

The importance of network latency:

Latency is a measure of delay. Network latency is the time needed to get from a source to a destination across some network infrastructure. A distributed approach can have an impact on the round-trip time of your network communications

Latency between Azure resources:

The goal here is to minimize the network latency between each layer of the application. How this is solved depends on your application and data architecture, but Azure provides mechanisms to solve this on several services

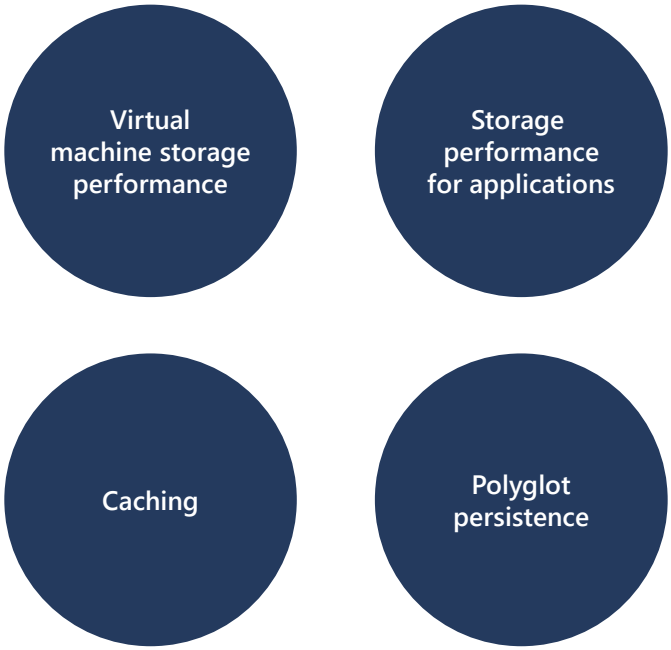
SSO with Azure Active Directory:

Latency between users and Azure resources can be improved by:

- Use a DNS load balancer for endpoint path optimization
- Use Azure Analysis Services as a Caching Service for Azure Data Warehouse
- Use CDN to cache content close to users
- Use ExpressRoute for connectivity from on-premises to Azure

49

Optimizing storage performance



50

Performance bottlenecks

Importance of requirements:

In theory, we could keep improving scalability and performance further and further without end. At some point, however, more improvement is prohibitively expensive, difficult, and doesn't have enough business impact

DevOps & application performance:

The idea behind DevOps is that we don't have development and infrastructure silos. Instead, they work together to effectively build, deploy, monitor, and maintain apps in streamlined process

Performance monitoring options in Azure

Azure monitor:

Azure Monitor provides a single management point for infrastructure-level logs and monitoring for most of your Azure services

Log Analytics:

With Log Analytics you can query and aggregate data across logs. This cross-source correlation can help you identify issues or performance problems that may not be evident when looking at logs or metrics individually

Application performance management:

Telemetry can include individual page request times, exceptions within your application, and even custom metrics to track business logic. This telemetry can provide a wealth of insight into apps

51

High availability

A highly-available service absorbs fluctuations in availability, load, and temporary failures in dependent services and hardware. The application remains online and available

Determine the service-level agreement of your application:

A service-level agreement (SLA) is an agreement between a service provider and a service consumer in which the service provider commits to a standard of service based on measurable metrics and defined responsibilities

Evaluate the HA capabilities of the application:

To evaluate the HA capabilities of your application, perform a failure analysis. Focus on single points of failure and critical components that would have a large impact on the application if they were unreachable, misconfigured, or started behaving unexpectedly

Evaluate the HA capabilities of dependent applications:

You'll need to understand not only your application's SLA requirements to your consumer, but also the provided SLAs of any resource that your application may depend on

52

Disaster recovery

Disaster recovery is about recovering from high-impact events that result in downtime and data loss

Create a disaster recovery plan:

A disaster recovery plan is a single document that details the procedures that are required to recover from data loss and downtime caused by a disaster and includes:

- Risk assessment and process inventory
- Recovery objectives
- Detailed recovery steps

Designing for Disaster recovery:

Disaster recovery is not an automatic feature. It must be designed, built, and tested, and should include the following:

- Data recovery and replication
- Process recovery
  - Azure Site Recovery
  - Service-specific features

Testing disaster recovery:

Disaster recovery planning doesn't end once you have a completed plan in hand. Testing the plan is a crucial aspect of disaster recovery, to ensure that the directions and explanations are clear and up-to-date

53

Backup and restore

Establish backup and restoration requirements

Azure backup and restore capabilities:

Azure Backup

Azure Blob storage

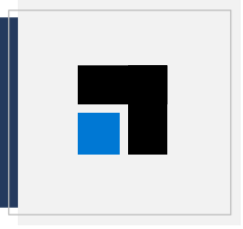
Azure SQL Database

Azure App Service

Verify backups and test restore procedures

54

Lab: Designing for scale and resiliency



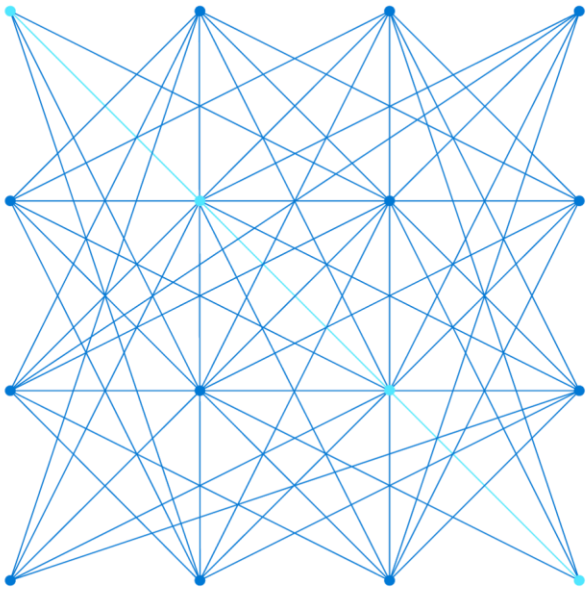
55

Microsoft Azure

56

Microsoft Azure

Module 06: Designing for efficiency and operations



57

Lesson objectives

- Maximize efficiency of cloud spend
- Use monitoring and analytics to gain operational insights
- Use automation to reduce effort and errors

58

Maximize your cloud spend

How the cloud changes your expenses

Track your cloud spend

Organize to optimize

Optimizing IaaS costs:

Compute

Right size virtual machines

Implement shutdown schedules for virtual machines

Apply compute cost discounts

Virtual machine disk storage cost optimization

Optimizing PaaS costs:

Optimizing Azure SQL Database costs

Optimizing Blob storage costs

Leverage consumption pricing models

Pause Compute Operations

59

Monitoring and analytics to gain operational insights

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

Core monitoring:

Core monitoring provides fundamental, required monitoring across Azure resources and gives you visibility into four key core monitoring areas:

- Activity logging
- Health of cloud services
- Metrics and diagnostics
- Recommendations on best practices

Deep infrastructure monitoring:

For typical IaaS workloads, there's more metrics and diagnostic information to gather from the network or operating systems. Pulling information from SQL Server to ensure it's properly configured, to analyzing free disk space across all the servers in your environment are all examples where Log Analytics can provide deep insights

Deep application monitoring:

You can take your monitoring capabilities even further by looking deep into your applications to identify performance issues, usage trends, and overall availability of services. By using an application performance management tool, you can better detect and diagnose issues that occur within your web apps and services

60

Automation to reduce effort and error

Managing the infrastructure of any type of workload involves configuration tasks. This configuration can be done manually, but it can be labor-intensive, error prone, and inefficient

Infrastructure as code:
Infrastructure as code is the management of infrastructure (networks, virtual machines, load balancers, and connection topology) in a descriptive model, using a versioning system similar to what is used for source code. there are two different approaches you can take: Imperative and declarative automation

Automation of operational tasks:
There are ongoing operational activities that can also be automated. Automating these tasks with Azure Automation reduces manual workloads, enables configuration and update management of compute resources, and centralizes shared resources such as schedules, credentials, and certificates

Automating development environments:
At the other end of the pipeline of your cloud infrastructure are the development machines used by developers to write the applications and services that are the core of your business. You can use Azure DevTest Labs to stamp out VMs with all of the correct tools and repositories that they need

61

Lab: Designing for efficiency and operations

62


Microsoft Azure
© Copyright Microsoft Corporation. All rights reserved.

63