Microsoft Azure

Microsoft Azure Well-Architected

Build and manage high-performing workloads

Agenda

- Why is being Well-Architected important?
- Overview: Microsoft Azure Well-Architected
- Overcoming workload quality inhibitors
- How to get started with the Well-Architecture Framework
- Resources

Data breaches cost you —and your customers

Customers' PII was the most frequently, and costliest compromised type of record per latest data breach study*

$3.86M	Average total cost of a data breach
80%	Number of breaches carried out with customer PII
$150	Customer PII average cost per record
$175	Increased cost per record of customer PII in breaches caused by a malicious attack
$137,000+	Remote workforce impact on average total cost of data breaches

Cost of a Data Breach Report 2020, IBM Security, Ponemon Institute

Run Well-Architected cloud workloads—
to create business value

Invest in these actions:

- Manage budget
- Improve workloads security
- Increase incident response
- Streamline internal processes
- Find costly mistakes
- Enhance workload performance

To avoid these consequences:

Expenses, losses

Trust

Damages

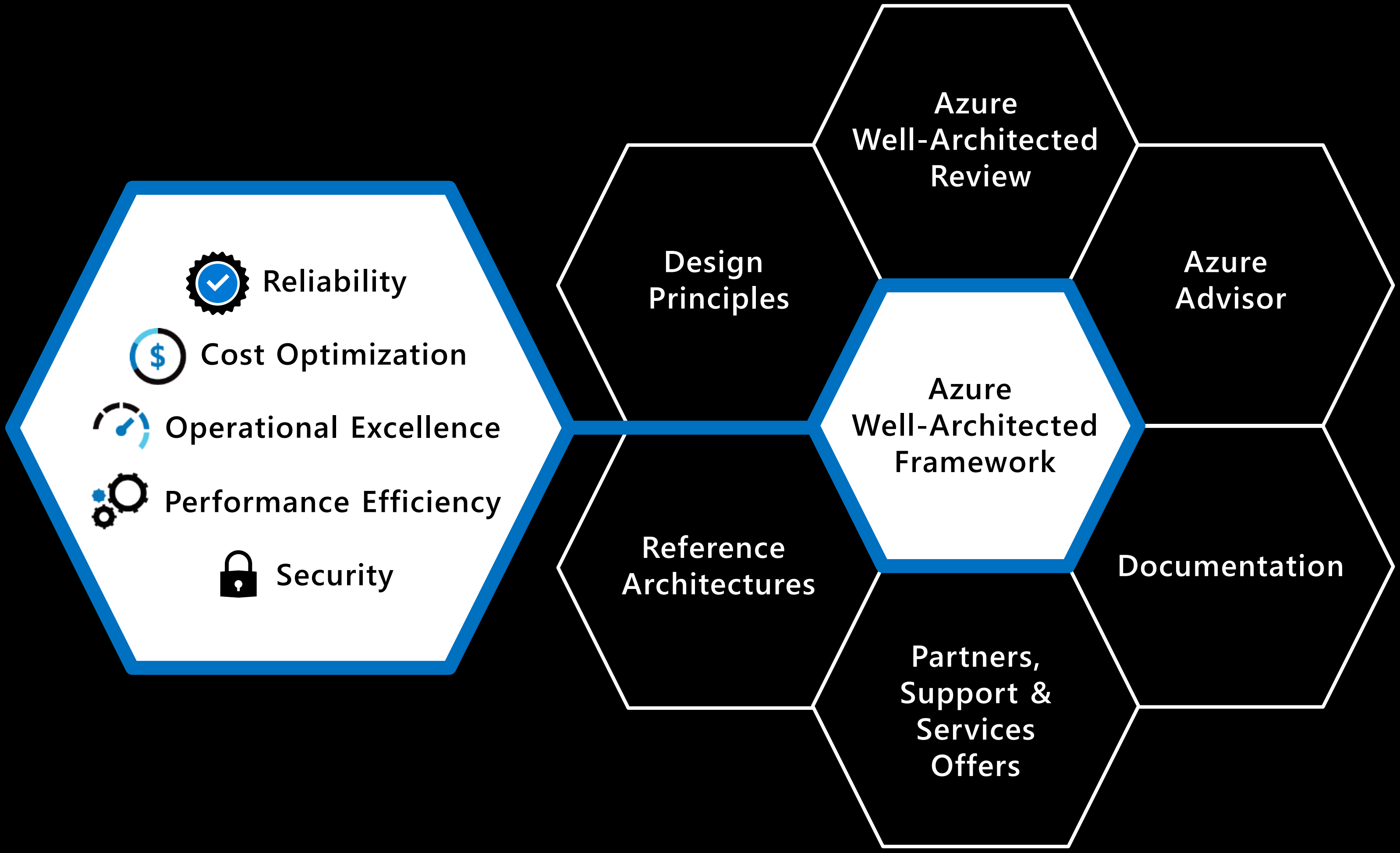
Azure Well-Architected

Build and manage high-performing workloads

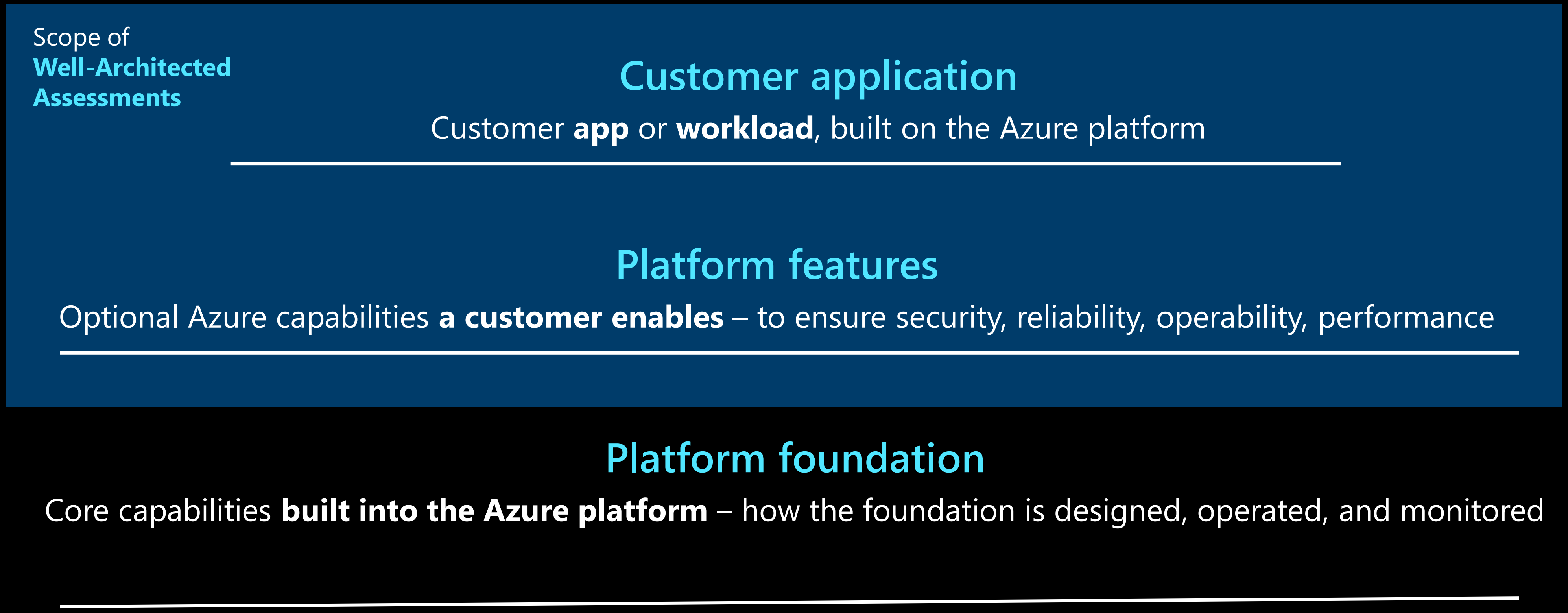
Build workloads with confidence with proven best practices

Design high-performing workloads using deep technical guidance

Optimize workloads with actionable focus areas



Building well-architected workloads is a **shared responsibility**



Scope of
Well-Architected Assessments

Customer application

Customer **app** or **workload**, built on the Azure platform

Platform features

Optional Azure capabilities **a customer enables** – to ensure security, reliability, operability, performance

Platform foundation

Core capabilities **built into the Azure platform** – how the foundation is designed, operated, and monitored

Microsoft Azure Well-Architected Framework

Architecture guidance and best practices, created for architects, developers and solution owners, to improve the quality of their workloads, based on 5 aligned and connected pillars

Cost Optimization

Operational Excellence

Performance Efficiency

Reliability

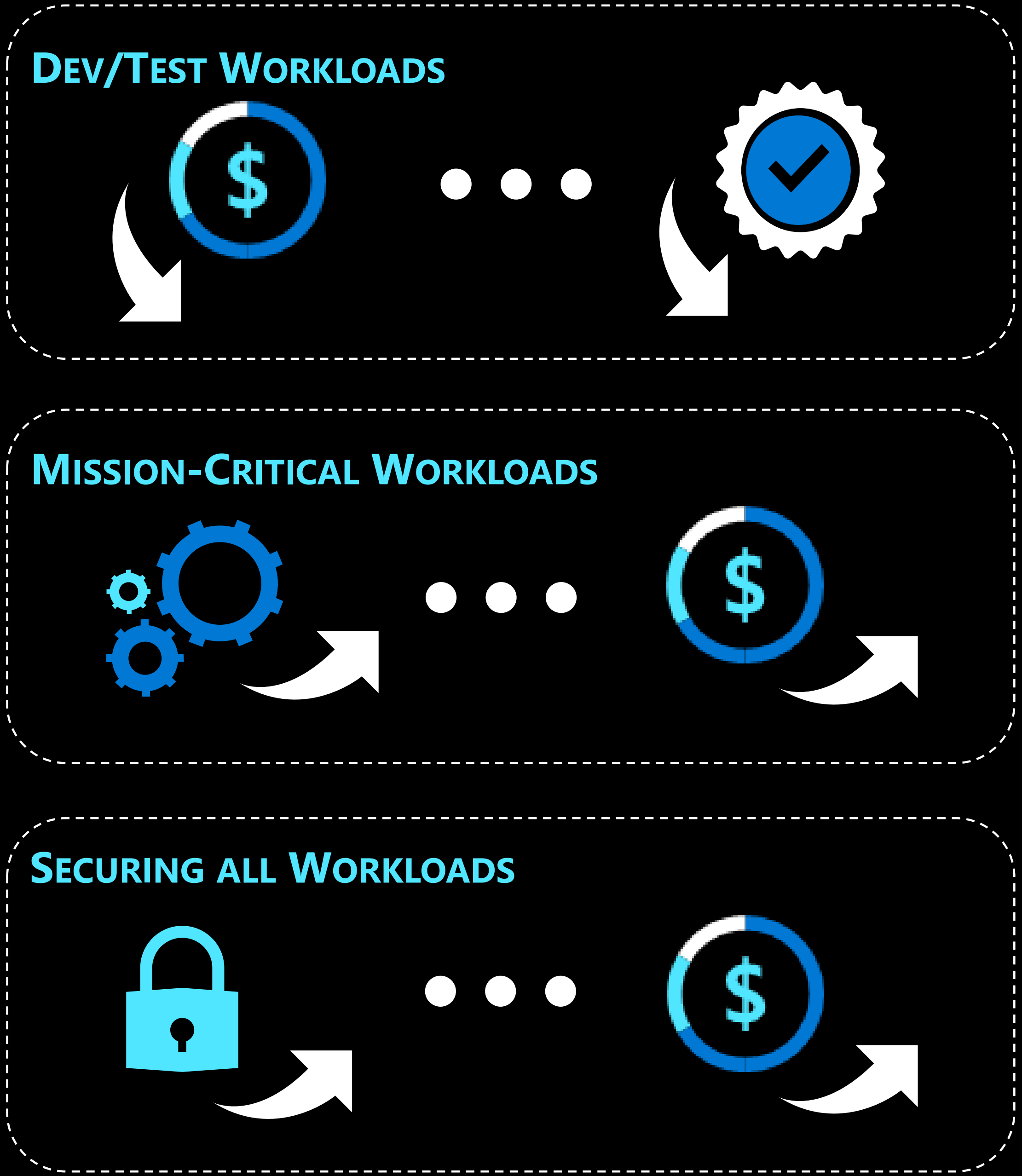
Security

https://aka.ms/wellarchitected/framework

It’s all about the trade-offs

Business requirements influence workload architecture decisions

https://aka.ms/wellarchitected/framework



DEV/TEST WORKLOADS

MISSION-CRITICAL WORKLOADS

SECURING ALL WORKLOADS

Overcoming workload quality inhibitors

Cost Optimization

- No cost and usage monitoring
- Unclear on underused or orphaned resources
- Lack of structure billing management
- Budget reductions due to lack of support for cloud adoption by LT/board

Operational Excellence

- Lack of rapid issue identification
- No deployment automation
- Absence of communication mechanisms and dashboards
- Unclear expectations and business outcomes
- No visibility on root cause for events

Performance Efficiency

- No monitoring new services
- No monitoring current workloads health
- No design for scaling
- Lack of rigor and guidance for technology and architecture selection

Reliability

- Unclear on resiliency features/capabilities for better architecture design
- Lack of data back up practices
- No monitoring current workloads health
- No resiliency testing
- No support for disaster recovery

Security

- No access control mechanism (authentication)
- No security threat detection mechanism
- Lack of security threat response plan
- No encryption process

https://aka.ms/wellarchitected/framework

Best practices to drive workload quality

Cost Optimization

- Azure Hybrid Benefit
- Reserve Instances
- Shutdown
- Resize
- Move to PAAS

Operational Excellence

- DevOps
- Deployment
- Monitor
- Processes and cadence

Performance Efficiency

- Design for scaling
- Monitor performance

Reliability

- Define requirements
- Test with simulations and forced failovers
- Deploy consistently
- Monitor health
- Respond to failure and disaster

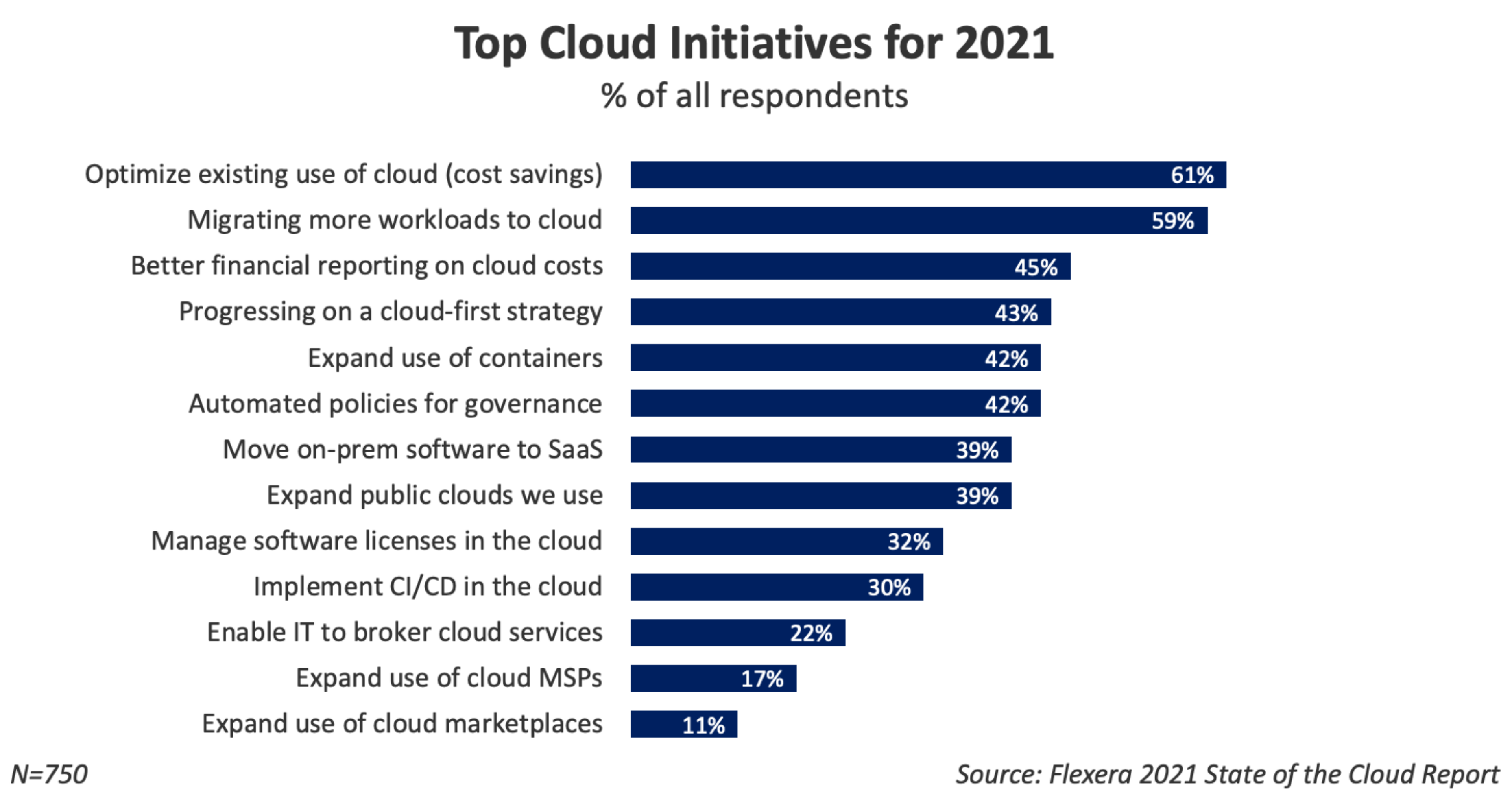
Security

- Identity and access management
- Infra protection
- App security
- Data encryption and sovereignty
- Security operations

https://aka.ms/wellarchitected/framework

Cost Optimization

Cost optimization =
top cloud initiative for the fifth year running



Source: Flexera 2021 State of the Cloud Report

Customer:
H&R Block

Industry:
Professional Services

Size:
10,000+ employees

Country:
United States

Products and services:
Microsoft Azure
Microsoft Azure Advisor
Microsoft Azure Cost Management and Billing
Microsoft Azure Well-Architected Framework

Read full story here

> “Our monthly spend year-over-year is nearly flat, while we now have approximately 30 percent more of our total compute in the cloud. Thanks to our partnership with Microsoft, our team has learned valuable techniques and strategies to continue optimizing our spend.”
>
> —Paul Clark, Director of Cloud, H&R Block

Situation:

As a leader in the modernization of the tax industry, H&R Block wanted to optimize its cloud infrastructure and provide better service for its customers.

Solution:

By engaging with its Microsoft account team and taking advantage of Azure Well-Architected Framework concepts, the company was able to optimize by replatforming to cloud-native services and modernizing its operating models.

Impact:

H&R Block is now equipped to take control of its monthly spend while also being able to move its total compute to the cloud and use its capabilities to benefit its business and customers.

Cost optimization categories

- Organizational
- Architectural
- Tactical

Optimize costs with tools, offers, and guidance

Cost optimization guidelines accelerating time to market, while avoiding capital-intensive solutions

Understand and forecast your costs

- Monitor your bill, set budgets, and allocate spending to teams and projects with Azure Cost Management + Billing
- Forecast costs for future investments with the Azure pricing and TCO calculator

Cost optimize your workloads

- Optimize your resources with Azure Advisor
- Follow best practices for workload design with the Azure Well-Architected Framework
- Save with Azure offers and licensing terms like the Azure Hybrid Benefit and Reservations

Control your costs

- Establish spending objectives and policies using the Microsoft Cloud Adoption Framework for Azure
- Implement cost controls in Azure Policy so your teams can go fast while complying with policy

Understand & forecast your cost

Principle: Monitor & optimize

Use alerts to monitor usage and spending

- Budget alerts notify you when spending reaches predetermined thresholds.
- Credit alerts notify you when your Azure Prepayment is consumed.
- Department spending quota alerts notify you when quotas are reached.

Auto-scaling policies provide cost savings

- When workloads are highly variable, choose smaller VM instances, then scale out, rather than up, to get the needed performance.
- Many applications can be made stateless, then auto-scaled for cost benefits.

Reserved instances can reduce costs

- Use Azure Reservations to lower costs by pre-paying for capacity.
- Analyze existing pay-as-you-go usage data in Azure Portal before opting into reserved instances.

Control your cost

Principle: Keep within cost constraints

Develop a cost model

- Map your organization's needs to specific offerings.
- Start with high-level requirements before considering design.
- Geographic and security decisions can have a huge impact on your costs.

Capture requirements

- Break down high-level goals into functional requirements.
- For each functional requirement, define metrics to estimate costs.

Cost tradeoffs

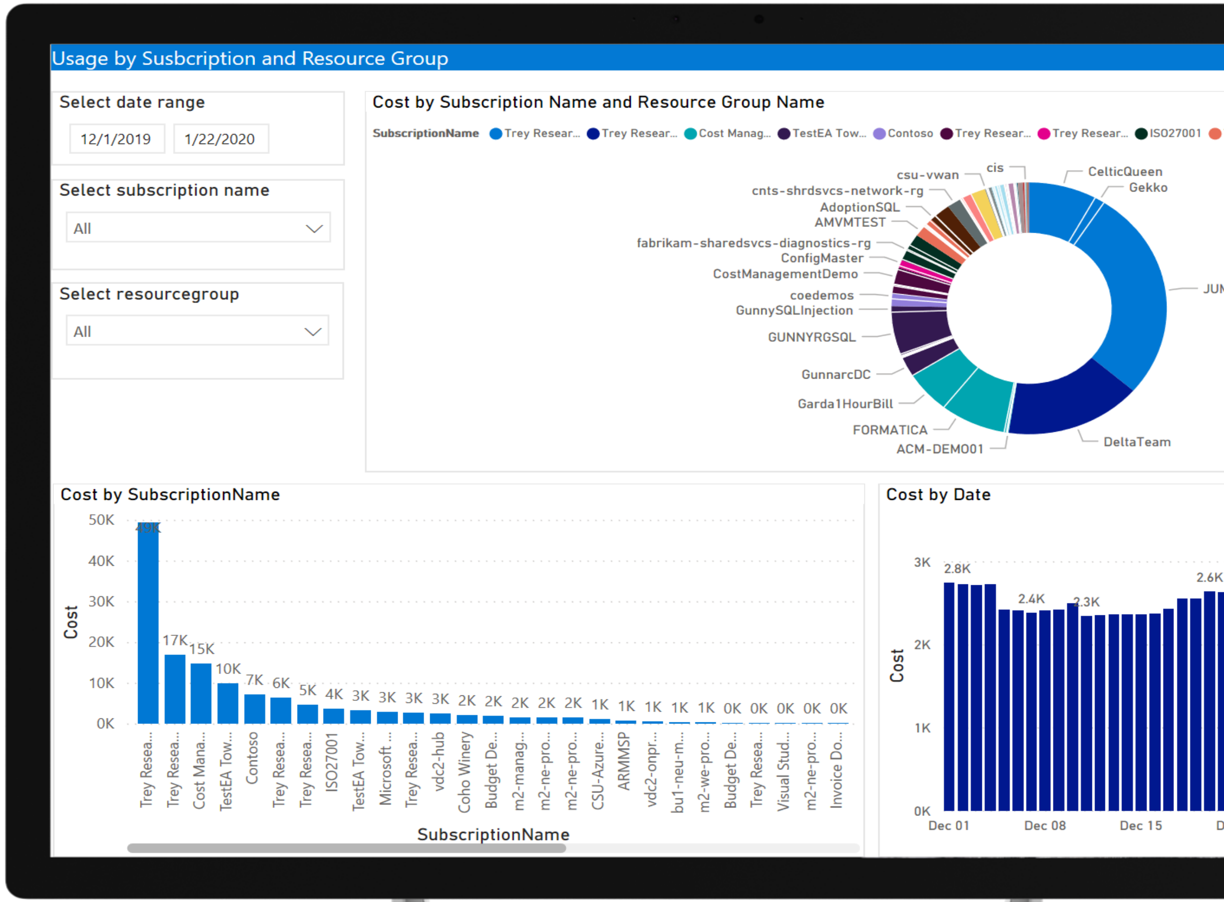
- Determine if the cost of high availability exceeds acceptable downtime.
- Increasing security of the workload will increase cost.
- Systems monitoring and automation might increase the cost initially but will reduce cost over time.

* Azure Cost Management + Billing Power BI App + RI Recommendations Dashboard

Reports included:

- Account overview
- Usage by Subscriptions and Resource Groups
- Top 5 Usage drivers
- Usage by Services
- Windows Server AHB Usage
- VM RI Coverage (shared recommendation)
- VM RI Coverage (single recommendation)
- RI Recommendations for all eligible services (Single/shared scope and 1Yr/3Yr terms)
- RI Savings
- RI Chargeback
- RI purchases
- Pricesheet

EA admin: run the ACM Power BI Template App and share the results with the Microsoft team

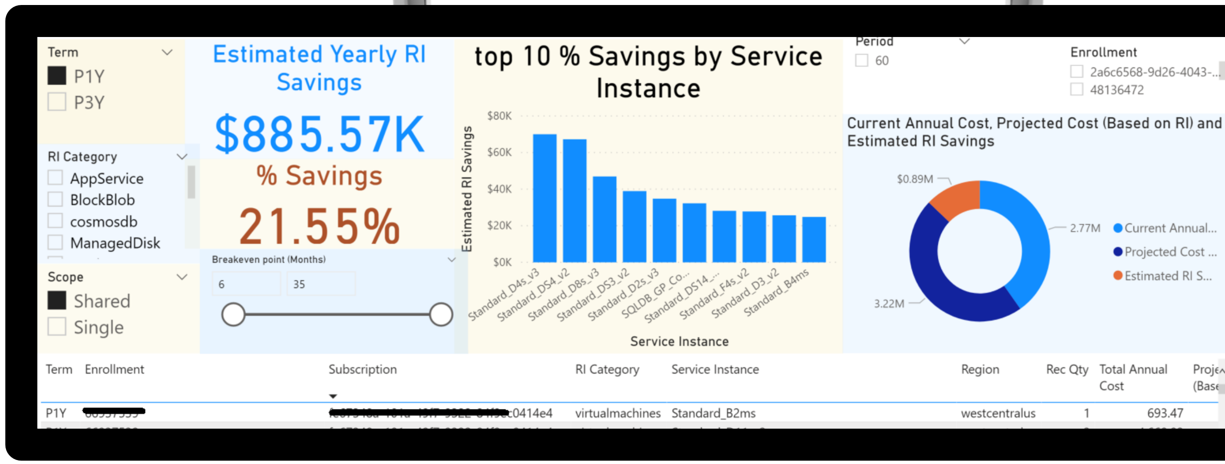


Usage by Susbcription and Resource Group

Select date range: 12/1/2019 – 1/22/2020

Select subscription name: All

Select resourcegroup: All



Term: ■ P1Y □ P3Y

RI Category: □ AppService □ BlockBlob □ cosmosdb □ ManagedDisk

Scope: ■ Shared □ Single

Estimated Yearly RI Savings

$885.57K

% Savings

21.55%

Breakeven point (Months): 6 – 35

Period: □ 60

Enrollment: □ 2a6c6568-9d26-4043-... □ 48136472

Term	Enrollment	Subscription	RI Category	Service Instance	Region	Rec Qty	Total Annual Cost	Proje (Base
P1Y	[redacted]	[redacted]0414e4	virtualmachines	Standard_B2ms	westcentralus	1	693.47	

Operational Excellence

Build, deploy, and manage workloads with trustworthy processes

Operational Excellence offers you the guidelines to create a sustainable application environment within building, deploying and maintaining workloads, while relying on automation, monitoring and testing.

Agile and Accurate Processes

- Apply DevOps to break down barriers between development and operations within the cloud journey.
- Reduce process risks by automating workloads with Azure Automation, Azure CLI and Azure PowerShell.
- Enjoy the flexibility of creating agile and independent workloads with Microservices.

Focused and assertive application monitoring

- Dive deep into your workloads' information with Log Analytics for infrastructure and with Azure Application Insights for application trends.
- Manage the health of your system and activity logging by consuming core monitoring insights provided by Azure Monitor.

Continuous Improvement

- Build and test workloads with Continuous Integration and Continuous Delivery (CI/CD) both in development and production stages.
- Perform extensive automated testing with Azure Pipelines or manual testing with Azure Testing Plans.

Agile and accurate processes

Principle: optimize build and release processes

From provisioning with Infrastructure as Code, to build and releases with CI/CD pipelines, to automated testing, embrace software engineering disciplines across your entire environment.

Infrastructure as Code

- Define the entire Infrastructure as Code just as you define your application
- Increase accuracy and reduce process risks preventing configuration drift
- Enable easy recreation of new environments, e.g., for developing new features

Continuous Integration & Continuous Delivery

- Build and test workloads with Continuous Integration and Continuous Delivery (CI/CD) both in development and production stages, to achieve a single and consistent way of building and deploying.
- Eliminate error-prone manual interventions
- Versioning of CI/CD pipelines for traceability of changes

Automated testing

- Perform extensive automated testing to ensure a stable code base and resource composition before deploying to critical systems
- Achieve a faster time-to-ship with fewer errors

Focused and assertive application monitoring

Principle: Monitor system and understand operational health

Implement systems and processes to monitor build and release processes, infrastructure health, and application health. Telemetry is critical to understanding the health of a workload and whether the service is meeting the business goals.

Monitor build and release processes

- Give developers early feedback on pushed code changes
- Avoid outages caused by the rollout of new features

Understand workload health to meet business goals

- Understand the business impact of reduced workload health
- Correlate events and metrics across different parts of your solution
- Respond to issues with self-healing capabilities

Monitor infrastructure and application health

- Build confidence in the overall health of your workload
- Dive deep into instrumentation with Log Analytics for infrastructure monitoring
- Instrument your code to collect all relevant events and metrics
- Use comprehensive dashboards that are tailored to your audiences
- Leverage Azure Application Insights for observing application trends

Continuous Improvement

Principle: Use loosely coupled architecture

Enable teams to independently test, deploy, and update their systems on demand without depending on other teams for support, services, resources, or approvals.

Strive for a true DevOps model

- Apply DevOps to break down barriers between development and operations within the cloud journey.
- Run agile and independent teams that are in charge of developing and running their parts of the workload
- Limit impact of issues by having clear boundaries between services

Microservices design

- Enjoy the flexibility of creating agile and independent workloads with microser vices.

Continuous Improvement

Principle: rehearse recovery and practice failure

Run DR drills on a regular cadence and use chaos engineering practices to identify and remediate weak points in application reliability. Regular rehearsal of failure will validate the effectiveness of recovery processes and ensure teams are familiar with their responsibilities.

Rehearse recovery

- Only tested recovery procedures will work in times of emergency
- Validate operation runbooks
- Run regular tests and conduct dry runs of failover scenarios

Practice failure

- Test your workload with injected faults in a safe environment
- Use Chaos Engineering practices to reach higher levels of maturity
- Employ a “Red Team” to find issues and weak points

Continuous Improvement

Principle: embrace operational improvements

Continuously evaluate and refine operational procedures and tasks, while striving to reduce complexity and ambiguity. This approach enables an organization to evolve processes over time, optimizing inefficiencies and learning from failures.

Evolve processes

- Having well-defined owners and playbooks for procedures and tasks is vital to optimizing operational effectiveness.
- Testing operational procedures and tasks should occur according to a reasonably regular cadence.
- Review operational incidents to improve operational effectiveness.
- Establish a Root Cause Analysis processes.

Optimizing inefficiencies through automation

- Save time, reduce risks and avoid errors by automating operational tasks or any deployments that may occur on a schedule, as a response to an event or monitoring alert, or ad-hock based on external factors.
- Automate deployments Infrastructure as Code to define the infrastructure that needs to be deployed.
- Optimize your workload configuration by automating software installs, adding data to a database, updating networking and other actions.

Performance Efficiency

Build and manage scalable and efficient workloads

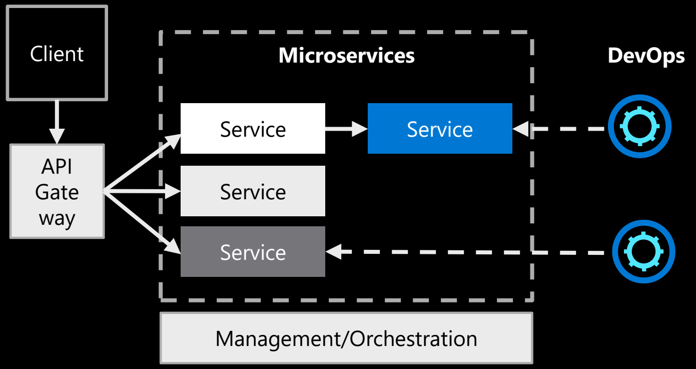
Performance Efficiency offers you the guidelines to design and manage workloads that scale according to load changes, and to design efficient systems, monitor processes, and optimize resources

Optimal service execution

- Manage resource scaling with Azure SQL Database and Azure App Services, or automatically with Azure Autoscale
- Optimize your network and storage with Azure Cosmos DB, Azure Traffic Manager and Azure Cache for Redis

Efficient architecture trade-offs

- Design parts of the process to be discrete and decomposable, to maximize compute resources and consider microservices architecture



Active response to performance issues

- Evaluate workload health levels with Azure Monitor and Log Analytics, to provision resources dynamically and scale based on demand
- Assess and remediate deep application performance issues and trends with Azure Application Insights
- Embrace a data-driven culture to deliver timely insights to everyone in your organization across all your data

Optimal service execution

Principle: Performance Testing and Capacity Planning

Load testing in pre-production provides insights and evidence on how your workload would perform at various scales, predicting when and how it could fail, so you can identify and fix errors

Test continuously

- Establish baselines for your application and its supporting infrastructure.
- Whenever code or infrastructure changes are made, always test the effect on performance.
- Knowing typical and peak system loads help you understand when it is operating outside of designed limits.

Anticipate load fluctuations

- Test for expected loads due to planned events like sales promotions or holidays.
- Plan for unexpected political, economic, and weather events.
- Choose paired regions and ensure that all regions can adequately scale to maximize uptime.

Carefully evaluate services and costs

- Review the Service Level Agreements (SLAs) of similar services to determine which is the best fit for your application.
- Consider business requirements impact when making trade-offs between cost and performance.
- Use cost calculators to estimate the initial and operational costs.

Efficient architecture tradeoffs

Principle: Distributed Architecture

Distributed architectures are complex and can require many areas of expertise and as much telemetry captured throughout the application and across all services as possible.

Distributed systems require more effort

- You must consider each application, its supporting services, and the latency between application layers.
- Ensure that all services can scale to support loads and that one service will not be a bottleneck.
- Services may need to scale differently under loads.

Test and tune Performance

- Define a service level objective (SLO) that defines performance targets for latency, number of requests, and exception rate for each workload.
- Use proven practices such as properly-instrumenting code, monitoring multiple load percentages, and systemic troubleshooting.

Avoid performance anti-patterns

- Performance antipatterns are common defective processes and implementations within organizations that are likely to cause scalability problems when an application is under pressure.
- Antipatterns may be obvious, such as an inability to scale from on-premises to the cloud.

Active response to performance issues

Principle: Monitoring

Define a comprehensive monitoring strategy to consider scalability, resiliency and performance. Use application telemetry and profiling to better identify issues, and Azure Data Explorer and Grafana to identify performance trends captured over time. Additional tools to make use of are …

Azure Monitor

- A comprehensive solution for collecting, analyzing, and acting on telemetry from your cloud and on-premises environments.
- Helps you maximize the availability and performance of your applications and services.

Log Analytics

- Edit and run log queries from data collected by Azure Monitor Logs and interactively analyze the results.
- Retrieve records matching particular criteria, identify trends, analyze patterns, and provide a variety of insights into your data.

Application Insights

- Assists you in understanding how your app is performing and how it's being used.
- Monitors a variety of data including request, response and failure rates, exceptions, page views and load performance.

Reliability

Building reliable applications in the cloud—

Enable systems to recover from failures and continue to function. In the cloud, we acknowledge up front that failures will happen. The goal is to minimize the effects of a single failing component, instead of trying to prevent overall failure.

Design for reliability

- Use Availability Zones where applicable to improve reliability and optimize costs.
- Design applications to operate when impacted by failures.
- Use the native resiliency capabilities of PaaS to support overall app reliability.
- Validate that required capacity is within Azure service scale limits and quotas.

Test for availability & resiliency

- Test regularly to validate existing thresholds, targets and assumptions.
- Verify how the end-to-end workload performs under failure conditions.
- Conduct load testing with expected peak volumes to test scalability and performance under load.
- Perform chaos testing by injecting faults.

Monitoring & diagnostics

- Define alerts that are actionable and effectively prioritized.
- Create alerts that poll for services nearing their limits and quotas.
- Use application instrumentation to detect and resolve performance anomalies.
- Troubleshoot issues to gain an overall view of application health.

Push Doctor

Customer:
Push Doctor
Industry:
Professional Services
Size:
50-999 employees
Country:
United Kingdom
Products and services:
Microsoft Azure
Microsoft Azure App Service
Microsoft Azure Application Gateway
Microsoft Azure Availability Zones
Microsoft Azure Monitor
Microsoft Azure Service Bus
Microsoft Azure SQL Database
Microsoft Power BI

Read full story here

Microsoft

"We've used Azure to build a resilient platform and help countless people get quick and easy healthcare access they can count on."

— Paul Smith, Enterprise Architect, Push Doctor

Situation:

Push Doctor, a patient/doctor video consultation platform based in the United Kingdom, needed highly available and scalable infrastructure that would provide the reliability its patients need to access remote healthcare support on their terms.

Solution:

Using Microsoft Azure platform as a service resources such as Azure App Service, Push Doctor's platform is now instantly scalable and highly secure, with an impressive 99.99 percent uptime. And thanks to duplicated workloads, it can seamlessly manage failovers.

Impact:

Push Doctor can now match patients with a general practitioner in a matter of hours, helping to potentially save the lives of people who would have otherwise waited much longer for a consultation—a service that has proved invaluable during the COVID-19 crisis.

Design for Reliability

Principle: Design applications to be resistant to failures

Use Availability Zones within a region

- If greater failure isolation than Availability Zones alone can offer, you should consider deploying to multiple regions.
- Multiple regions should be used for failover purposes in a disaster state.
- Additional costs—data, networking and the Azure Site Recovery service—should be considered.

Respond to failure

- Resilient application architectures should be designed to recover gracefully from failures in alignment with defined reliability targets.
- Define an availability strategy to capture how the application remains available when in a failure state.
- Define a Business Continuity Disaster Recovery strategy for the application and/or its key scenarios.

Considerations for improving reliability

- Use Platform as a Service (PaaS), which offers native resiliency capabilities to support overall application reliability.
- Design your application to automatically scale in and out.
- Review Azure subscription and service limits to validate that required capacity is within quotas.

Test for availability and resiliency

Principle: Define, automate, and test operational processes

Testing, testing, testing

- Simulation Testing involves creating real-life situations and demonstrates the effectiveness of proposed solutions.
- Use Fault Injection Testing to check the system resiliency during failures—by triggering failures or by simulating them.
- Load Testing is crucial for identifying failures that only happen under load, (e.g., an overwhelmed back-end database, or service throttling).

Respond to failure

- Resilient application architectures should be designed to recover gracefully from failures in alignment with defined reliability targets.
- Define an availability strategy to capture how the application remains available when in a failure state.
- Define a Business Continuity Disaster Recovery strategy for the application and/or its key scenarios.

Backup & disaster recovery

- Create and fully test a disaster recovery plan using the actual resources needed to restore functionality.
- Perform an operational readiness test for failover to the secondary region and for failback to the primary region.
- Codify the steps required to recover or failover to a secondary region to limit the impact of an outage.

Monitoring application health

***P**rinciple: Define, automate, and test operational processes*

Alerts & dashboards

- Azure Service Health provides a view into the health of Azure services and regions, as well as communications about outages and planned maintenance activities.
- Azure Resource Health provides information about the health of individual and is highly useful when diagnosing unavailable resources.
- Azure dashboards provides a consolidated view of data from Application Insights, Log Analytics, Azure Monitor metrics, and Service Health.

Subscription & service limits

- If your application requires more storage accounts than are currently available in your subscription, create a new subscription with additional storage accounts.
- Identify scalability targets for VMs including VM size, number of disks, CPU, and memory.
- To avoid data throttling, review your Azure SQL Database requirements to ensure that they are adequate.

Backup & disaster recovery

- Create and fully test a disaster recovery plan using the actual resources needed to restore functionality.
- Perform an operational readiness test for failover to the secondary region and for failback to the primary region.
- Codify the steps required to recover or failover to a secondary region to limit the impact of an outage.

Security

Build and manage proactively secured workloads

Security offers you the guidelines to protect, detect, and respond to threats across your Azure environment.

Built on a secure foundation

- Design assuming workload failure with multi-layer protection controls.
- Build workloads using zero-trust principles in both IaaS and PaaS.
- Deploy Azure's security investments, resources, and compliance certifications.

Proactively stay secure with native controls

- Azure Security Center continuously manages workload security from a single Azure Web dashboard
- Azure Web Application Firewall (WAF) provides centralized protection of your web applications from common exploits and vulnerabilities
- Manage identity and access for your workload with Azure Active Directory

Detect and respond to threats

- Microsoft Security Graph, collects and streamlines secure and scalable delivery of Microsoft Graph data sources for Azure development tools to build intelligent applications
- Azure Defender leverages automated workflows to deploy threat protection for workload
- Azure Sentinel identify and mitigate threats against your workloads

Built on a secure foundation

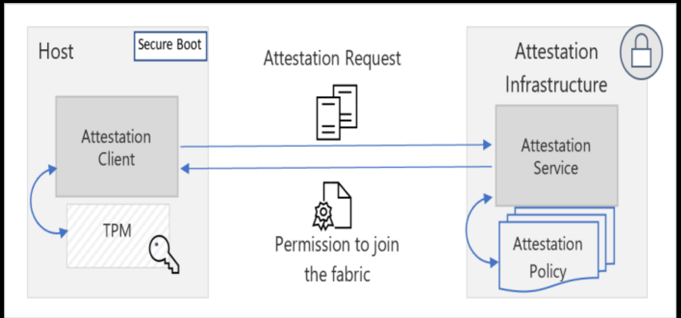
Principle: Build a comprehensive strategy

Protect customer data

- Azure Active Directory manages access to Azure resources.
- Azure Key Vault stores sensitive data such as certificates, connection strings, and tokens.
- Azure Security Benchmark provides recommendations to improve the security of your workloads, data, and services.

Secure hardware

- Azure is hosted on custom-built hardware with integrated security.
- Host Attestation Service ensures host machines are trust-worthy before allowed to interact with customer data.



Host attestation service high-level architecture

Test and monitor

- Run simulated penetration attacks to detect system vulnerabilities and validate defenses.
- Classify, protect, and monitor sensitive data assets using access control, encryption, and logging.

Built on a secure foundation

Principle: Assume Zero Trust

DDoS protection

DDOS protection tuned to your application traffic patterns

Web Application Firewall

Centralized inbound web application protection from common exploits and vulnerabilities

Azure Firewall

Data exfiltration protection with centralized outbound & inbound (non-HTTP/S) network & application (L3-L7) filtering

Network Security Groups

Distributed inbound & outbound network (L3-L4) traffic filtering on VM, Container or subnet

VNET Integration

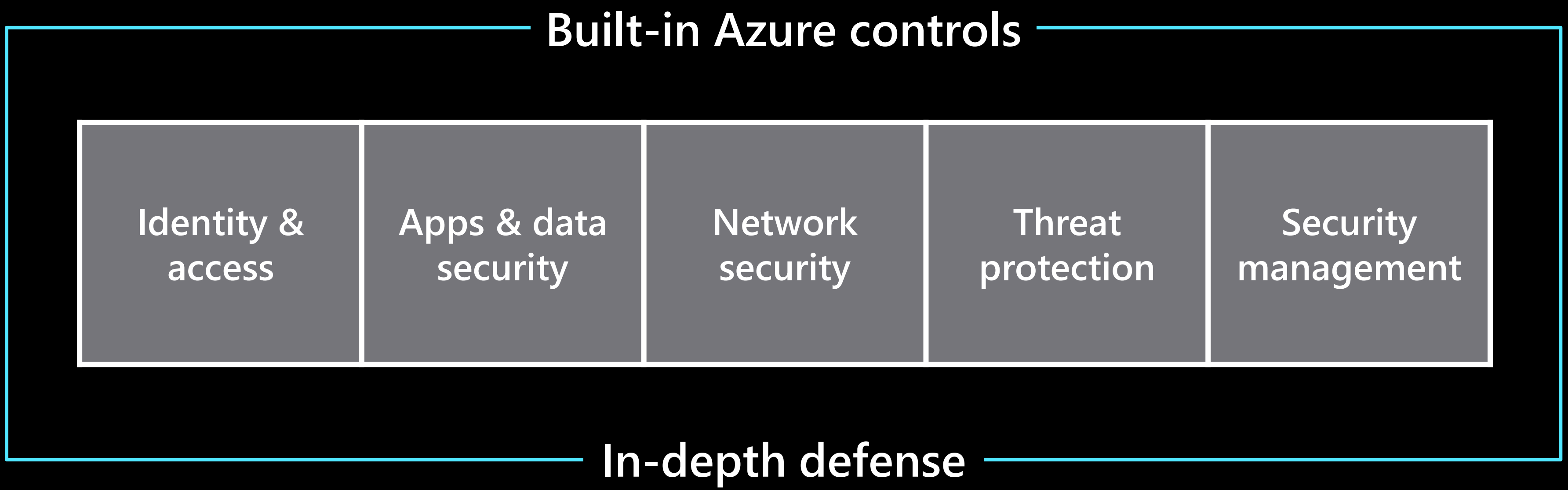
Virtual network-restricted access to Azure service resources (PaaS)

Application protection

Segmentation

Proactively stay secure with native controls (1 of 2)

Principle: Leverage native controls



Built-in Azure controls

Identity & access

Apps & data security

Network security

Threat protection

Security management

In-depth defense

Eliminate and reduce effort required to integrate external security tooling, and update integrations with service provider-supported native controls

Proactively stay secure with native controls (2 of 2)

Principle: Leverage native controls

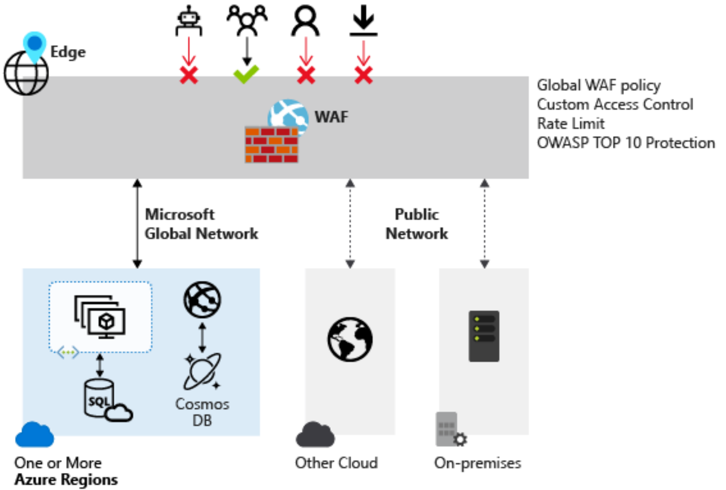
Azure Security Center

Unified infrastructure security management system:

- Strengthens the security posture of your data centers.
- Provides advanced threat protection across hybrid workloads in the cloud—on Azure, or on-premises.

Web Application Firewall

- Centralized protection and inspection of HTTP requests to prevent attacks such as SQL Injection or Cross-Site Scripting.



Azure Active Directory

- AAD's cloud-based identity and access management service helps organizations to securely access resources across environments.
- Managed Identities eliminates the need to store credentials that could be leaked.
- Azure AD Connect synchronizes Azure AD with an existing on-premises directory.

Detect and respond to threats

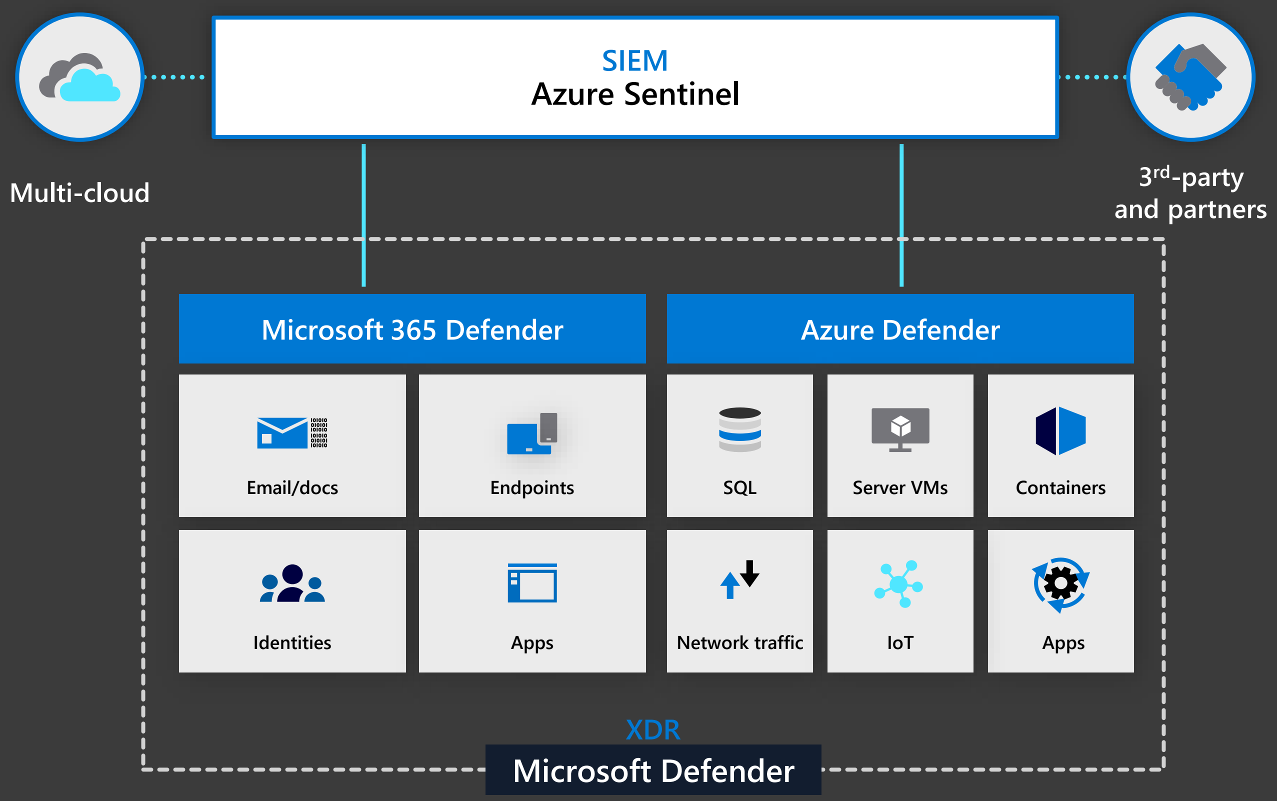
Principle: Design for Resilience

Native security & governance

- ASC/Secure Score
- Firewall
- Web App Firewall
- SQL Protection
- API Protection

•••

Native threat detection



SIEM
Azure Sentinel

Multi-cloud

3rd-party
and partners

Microsoft 365 Defender

Email/docs
Endpoints
Identities
Apps

Azure Defender

SQL
Server VMs
Containers
Network traffic
IoT
Apps

XDR
Microsoft Defender

How can you get started?

Optimize **existing** workloads

☑ Leverage Azure Advisor Score to identify optimization opportunities

☑ Understand changes needed or incidents occurred

☑ Review Well-Architecture Framework

☑ Consider architecture design trade offs to achieve business goals

☑ Define and implement recommendations

☑ Establish a regular cadence for workload optimization

Design & deploy **new** workloads

☑ Align workload architecture to business priorities

☑ Review Well-Architecture Framework

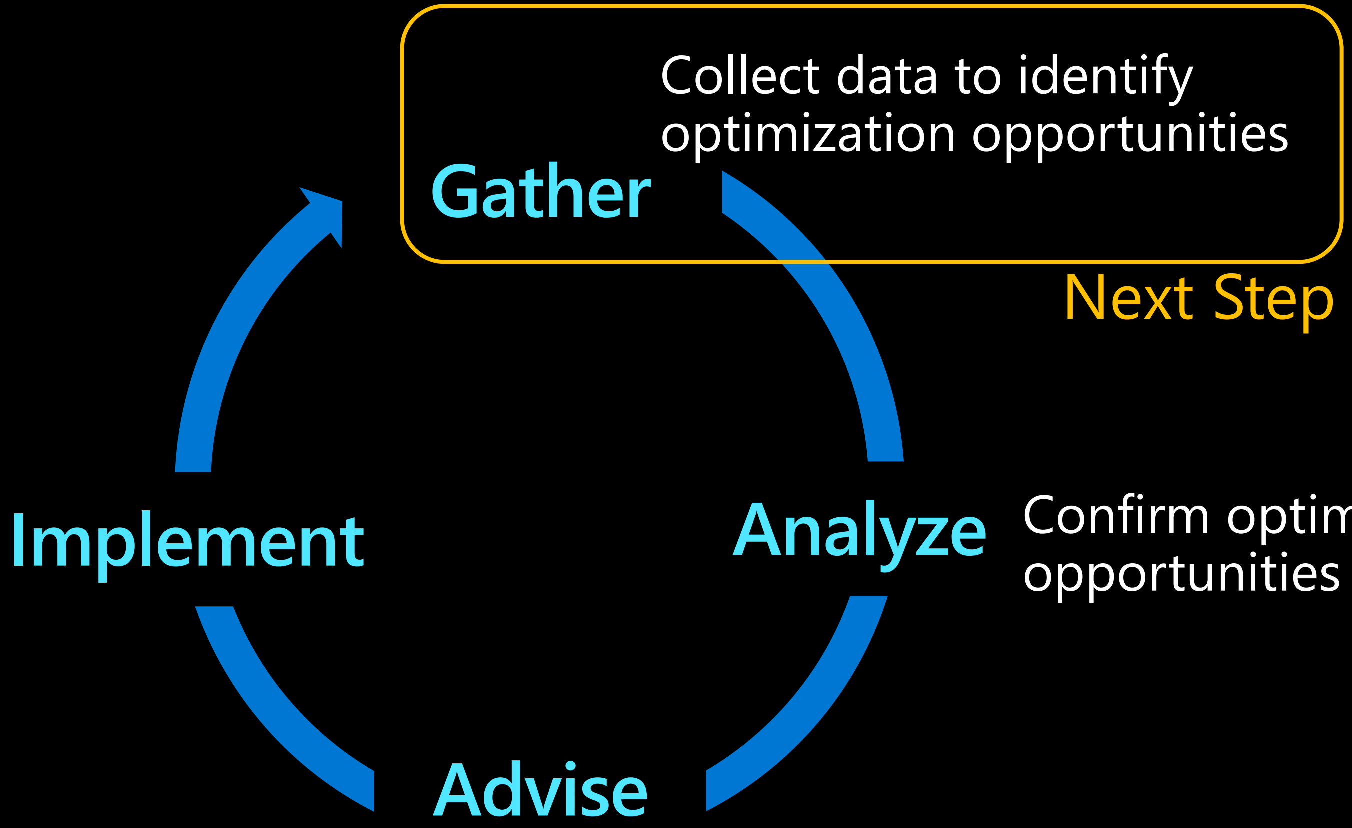
☑ Leverage the Azure Well-Architected Review to assess workload architecture design

☑ Consider architecture design trade offs to achieve business goals

☑ Build, deploy and manage workloads on Azure

Optimize existing workloads –

Engagement Overview Process



Gather

Collect data to identify optimization opportunities

Next Step

Analyze

Confirm optimization opportunities

Advise

Actionable plan and recommendations to optimize

Implement

Implement recommendations and continuous process

Technical Areas of Expertise Needed for Operational Excellence Workshop

Architecture

- Design
- Targets & Non-Functional Requirements
- Key Scenarios
- Dependencies
- Application Composition

Operational Procedures

- Recovery & Failover
- Scalability & Capacity Model
- Configuration & Secrets Management
- Operational Lifecycles
- Patch & Update Process (PNU)

Health Modelling

- Application-Level Monitoring
- Resource/Infrastructure Level Monitoring
- Data Interpretation & Health Modelling
- Dashboarding
- Alerting

Deployment & Testing

- Application Deployments
- Application Infrastructure Deployments & Infrastructure as Code (IaC)
- Build Environments
- Testing & Validation

Security & Compliance

- Identity and Access
- Security Center

Operational Model & DevOps

- General
- Roles & Responsibilities
- Common Engineering Criteria

Technical Areas of Expertise Needed in Performance Efficiency Workshop

Application Design

- Design Patterns
- Transactional
- Disaster Planning

Monitoring

- Logging
- Performance Targets

Capacity Planning

- Usage Prediction
- Service SKU
- Disaster Recovery

Performance Testing

- Resource & Performance Planning
- Load Capacity
- Test Coverage
- Benchmarking
- Tooling

Troubleshooting

- Operational Model
- DevOps

Technical Areas of Expertise Needed for Reliability Workshop

Architecture

- Design
- Targets & Non-Functional Requirements
- Key Scenarios
- Dependencies
- Application Composition

Operational Procedures

- Recovery & Failover
- Scalability & Capacity Model
- Configuration & Secrets Management
- Operational Lifecycles
- Patch & Update Process (PNU)

Health Modelling

- Application-Level Monitoring
- Resource/Infrastructure Level Monitoring
- Data Interpretation & Health Modelling
- Dashboarding
- Alerting

Deployment & Testing

- Application Deployments
- Application Infrastructure Deployments & Infrastructure as Code (IaC)
- Build Environments
- Testing & Validation

Security & Compliance

- Identity and Access
- Security Center

Operational Model & DevOps

- General
- Roles & Responsibilities
- Common Engineering Criteria

Technical Areas of Expertise Needed for Security Workshop

Architecture

- Plan & Design
- Business Outcomes
- Core, Workload, & Hybrid Dependencies
- Cloud Adoption Framework & Security Benchmarks
- Shared Responsibility

Security Governance

- Posture Management (Security Center)
- Policy, Standards, & Compliance (NIST, CIS, Industry)
- Data & Information Management
- Metrics, KPI, & Dashboarding

Innovation Security

- Application/Infrastructure Deployments
- DevOps, Infrastructure as Code (IaC), CI/CD
- Minimum Viable Security Product (ESLZ, WAF)

Security Operations

- Detect & Respond, Alert & Monitor (Sentinel)
- Threat Intelligence
- Proactive Threat Hunting
- Incident Management & Recovery
- Red Team / Blue Team Exercises

Asset Protection

- Asset Inventory, Patch & Update
- Network Isolation
- Firewall & Web Application Firewall, DDoS
- Encryption, Certificates, Keys, & Secrets
- Discovery & Classification

Access Control

- Access, Roles, & Permissions (RBAC, ABAC, Entitlements)
- Zero Trust Access Control (Conditional Access & MFA)
- Privileged Access

Customer Stakeholders (Example)

- ❑ Solution Architect
- ❑ Cloud Architect
- ❑ Operations Architect
- ❑ DevOps / SRE Lead
- ❑ SecOps Lead
- ❑ Solution Owner
- ❑ Solution Architect
- ❑ Cloud Architect
- ❑ Network Architect
- ❑ Data Architect
- ❑ Security Architect
- ❑ DevOps / SRE Lead
- ❑ Project Manager

Technical Workshop Approach

Microsoft-led review workshop lasting 1-to-2 days	Can be delivered as compressed virtual workshops spanning 2-to-4 hours
‘Top-down’ end-to-end assessment of the entire application and its design path	Covering all significant technical domains
‘Question & Answer’ format applied to solicit key data points	Relevant application context is required to identify risks and align associated recommendations

Key Outcomes

Identify key risks of the application, across prioritized well-architected pillars

Propose actionable and prioritized recommendations to address identified risks

- P0 – Critical short-term remediation
- P1 – Strongly recommended mid-term improvements
- P2 – Long-term sustainability recommendations

Capture key findings and associated recommendations in a Well-Architected report focused on the reviewed workload

Provide guidance for implementing critical short-term recommendations

Architect & optimize workloads for success

Leverage assessment
Azure Well-Architected Review
(aka.ms/wellarchitected/review)

Get trained
Well-Architected Learning Path
(aka.ms/wellarchitected/learn)

Browse Reference Architectures
Azure Architectures
(aka.ms/wellarchitected/referencearch)

Azure Enablement Show
Channel 9 Show
(aka.ms/azenable)

Review Design Principles
Well-Architected Design Principles
(aka.ms/wellarchitected/designprinciples)

Review the Documentation
Azure Well-Architected Framework
(aka.ms/wellarchitected/framework)

Engage a partner
Azure Partners

Find Service Offers
MS Consulting Services
(aka.ms/WAFServices)

Thank you!

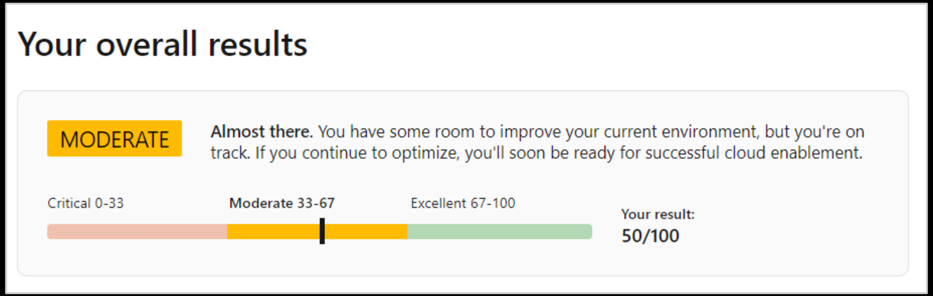
Additional Supporting slides

Well-Architected Review

Online Tool

Assess your workloads using the tenets found in the Microsoft Azure Well-Architected Framework:

- Understand the Well-Architected level of your workload environment.



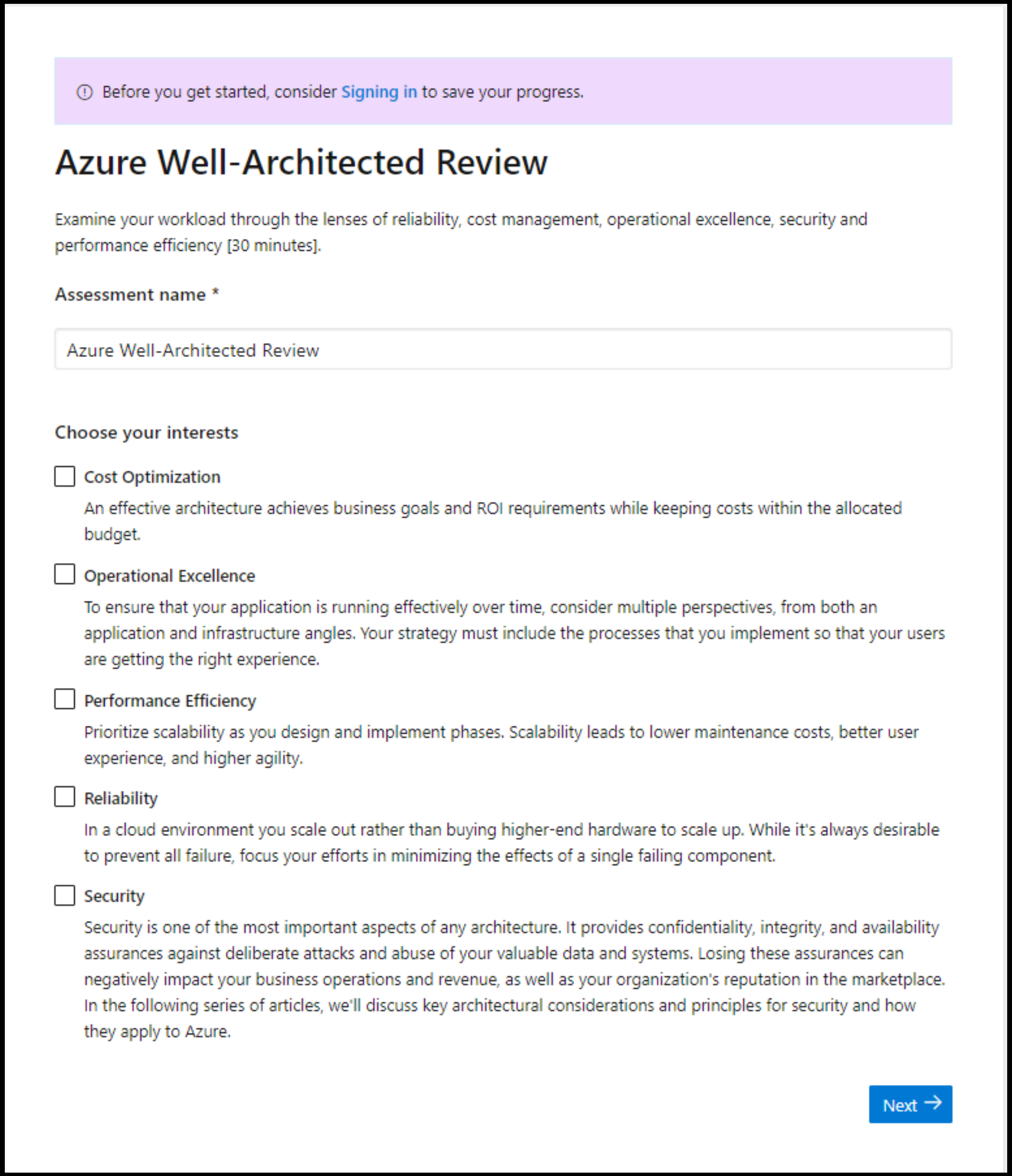
Your overall results

MODERATE **Almost there.** You have some room to improve your current environment, but you're on track. If you continue to optimize, you'll soon be ready for successful cloud enablement.

Your result: 50/100

- Access guidance for next steps of your workload improvement process.

aka.ms/wellarchitected/review



ⓘ Before you get started, consider Signing in to save your progress.

Azure Well-Architected Review

Examine your workload through the lenses of reliability, cost management, operational excellence, security and performance efficiency [30 minutes].

Assessment name *

Azure Well-Architected Review

Choose your interests

□ **Cost Optimization**

An effective architecture achieves business goals and ROI requirements while keeping costs within the allocated budget.

□ **Operational Excellence**

To ensure that your application is running effectively over time, consider multiple perspectives, from both an application and infrastructure angles. Your strategy must include the processes that you implement so that your users are getting the right experience.

□ **Performance Efficiency**

Prioritize scalability as you design and implement phases. Scalability leads to lower maintenance costs, better user experience, and higher agility.

□ **Reliability**

In a cloud environment you scale out rather than buying higher-end hardware to scale up. While it's always desirable to prevent all failure, focus your efforts in minimizing the effects of a single failing component.

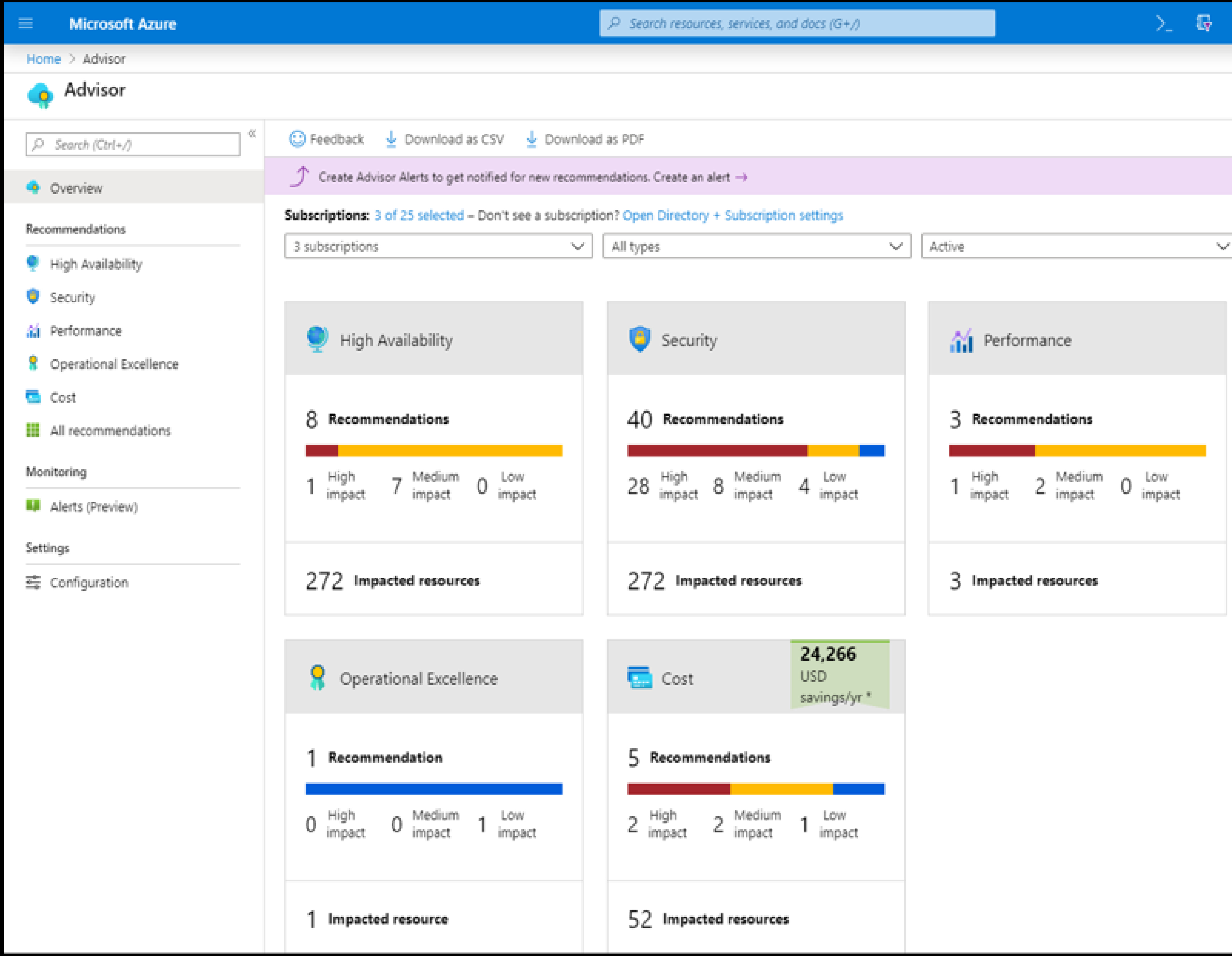
□ **Security**

Security is one of the most important aspects of any architecture. It provides confidentiality, integrity, and availability assurances against deliberate attacks and abuse of your valuable data and systems. Losing these assurances can negatively impact your business operations and revenue, as well as your organization's reputation in the marketplace. In the following series of articles, we'll discuss key architectural considerations and principles for security and how they apply to Azure.

Azure Advisor

Practical recommendations based on your usage and configurations, such as:

- Create Azure Service Health alerts.
- Delete and re-create your pool to remove a deprecated internal component.
- Repair invalid log alert rules.
- Design your storage accounts to prevent reaching the maximum subscription limit.



Create Advisor Alerts to get notified for new recommendations. Create an alert →

Subscriptions: 3 of 25 selected – Don't see a subscription? Open Directory + Subscription settings

3 subscriptions | All types | Active

High Availability: 8 Recommendations — 1 High impact, 7 Medium impact, 0 Low impact — 272 Impacted resources

Security: 40 Recommendations — 28 High impact, 8 Medium impact, 4 Low impact — 272 Impacted resources

Performance: 3 Recommendations — 1 High impact, 2 Medium impact, 0 Low impact — 3 Impacted resources

Operational Excellence: 1 Recommendation — 0 High impact, 0 Medium impact, 1 Low impact — 1 Impacted resource

Cost: 24,266 USD savings/yr * — 5 Recommendations — 2 High impact, 2 Medium impact, 1 Low impact — 52 Impacted resources

Azure.com: aka.ms/advisor

Tool: aka.ms/azureadvisor

Documentation: aka.ms/docs/advisor

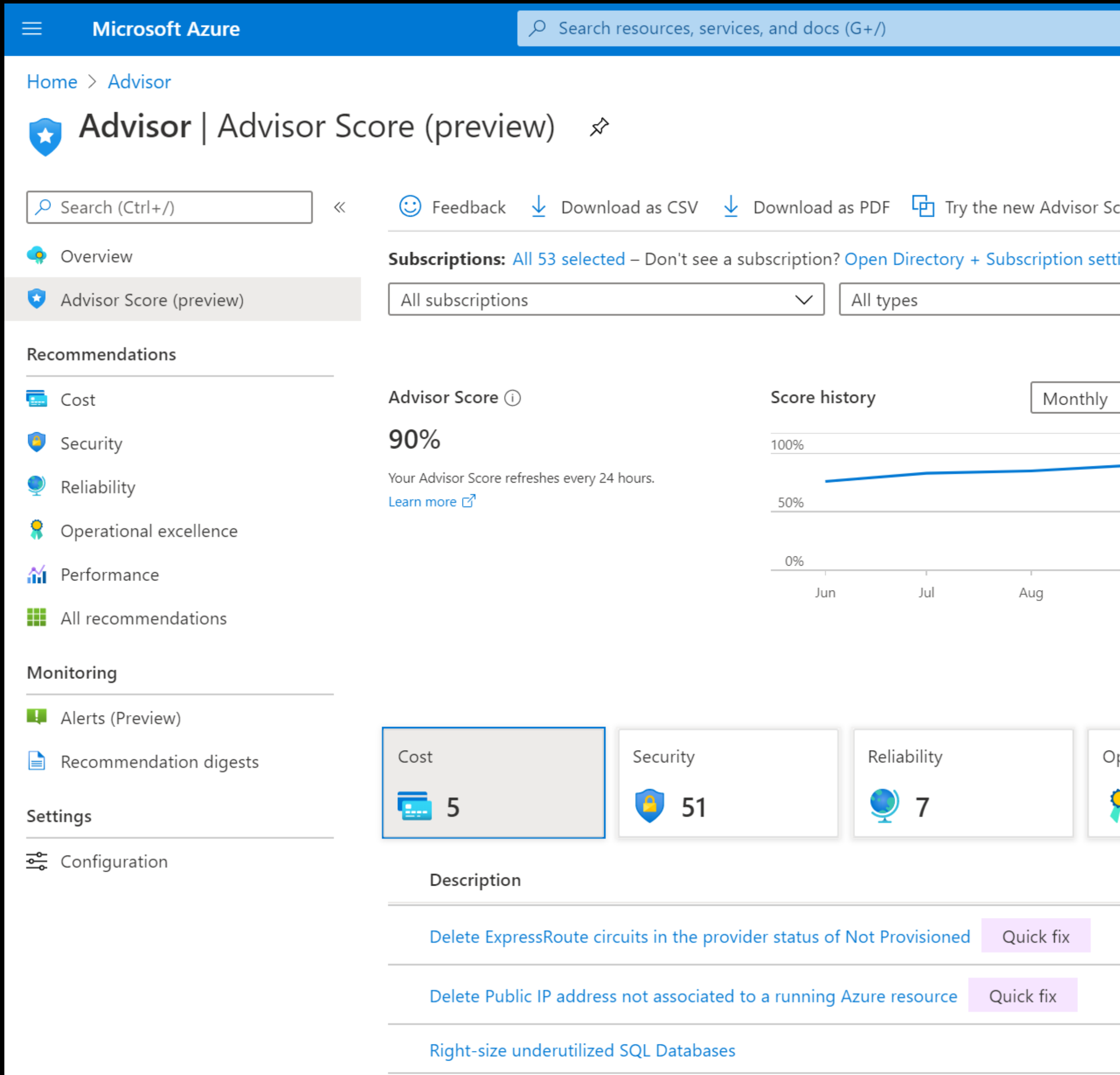
Advisor Score

A core feature of Azure Advisor that aggregates recommendations into a simple, actionable score.

Personalized and actionable best practice recommendations, including how to:

- Improve the posture of your workloads and optimize your Azure deployments.
- Proactively prevent top issues by following best practices.
- Assess your Azure workloads against the five pillars of the Microsoft Azure Well-Architected Framework.

Azure.com: aka.ms/advisor
Tool: aka.ms/azureadvisor
Documentation: aka.ms/docs/advisor



Microsoft Azure

Search resources, services, and docs (G+/)

Home > Advisor

Advisor | Advisor Score (preview)

Search (Ctrl+/)

Overview

Advisor Score (preview)

Recommendations

Cost

Security

Reliability

Operational excellence

Performance

All recommendations

Monitoring

Alerts (Preview)

Recommendation digests

Settings

Configuration

Feedback | Download as CSV | Download as PDF | Try the new Advisor Sc

Subscriptions: All 53 selected – Don't see a subscription? Open Directory + Subscription setti

All subscriptions | All types

Advisor Score

90%

Your Advisor Score refreshes every 24 hours.

Learn more

Score history | Monthly

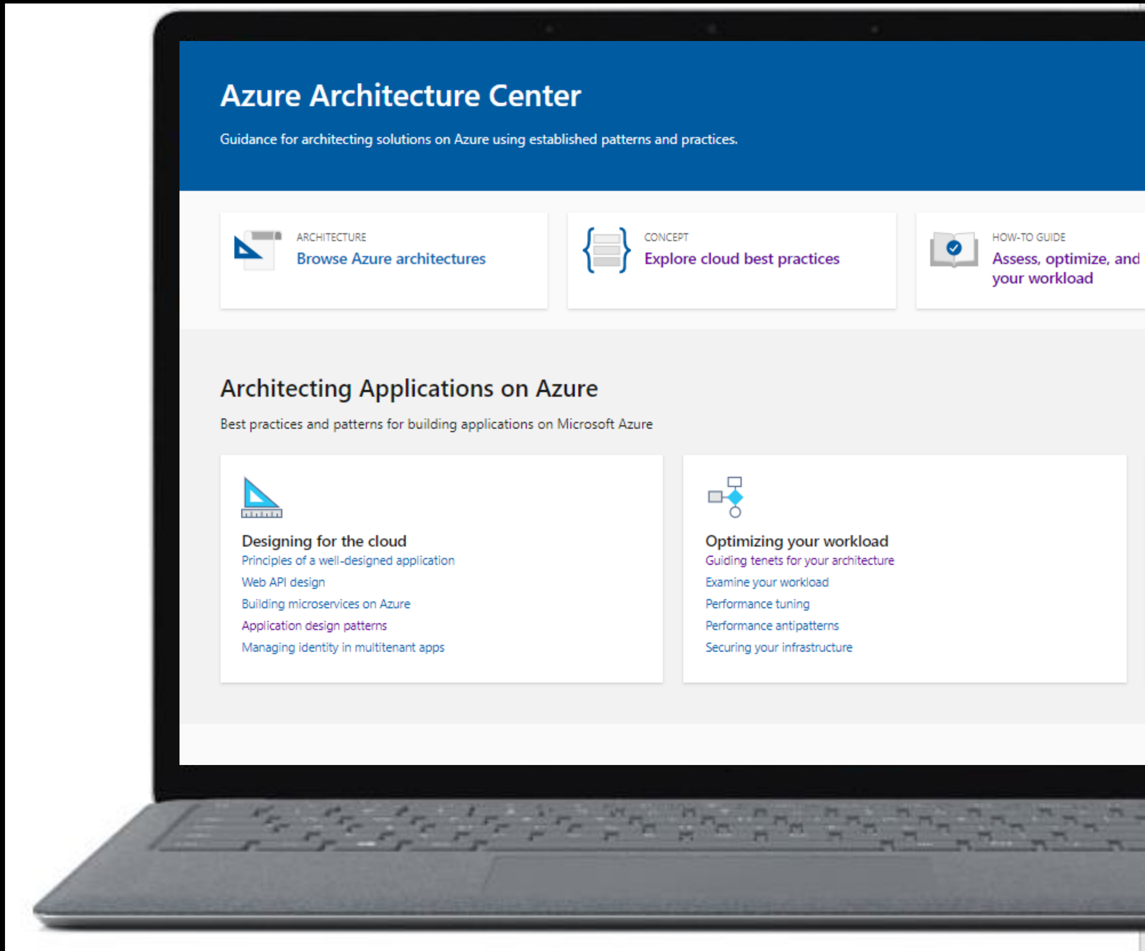
Cost	Security	Reliability	Op
5	51	7	

Description	
Delete ExpressRoute circuits in the provider status of Not Provisioned	Quick fix
Delete Public IP address not associated to a running Azure resource	Quick fix
Right-size underutilized SQL Databases	

Documentation

Microsoft Azure Architecture Center

https://aka.ms/architecture



Azure Architecture Center

Guidance for architecting solutions on Azure using established patterns and practices.

ARCHITECTURE
Browse Azure architectures

CONCEPT
Explore cloud best practices

HOW-TO GUIDE
Assess, optimize, and
your workload

Architecting Applications on Azure

Best practices and patterns for building applications on Microsoft Azure

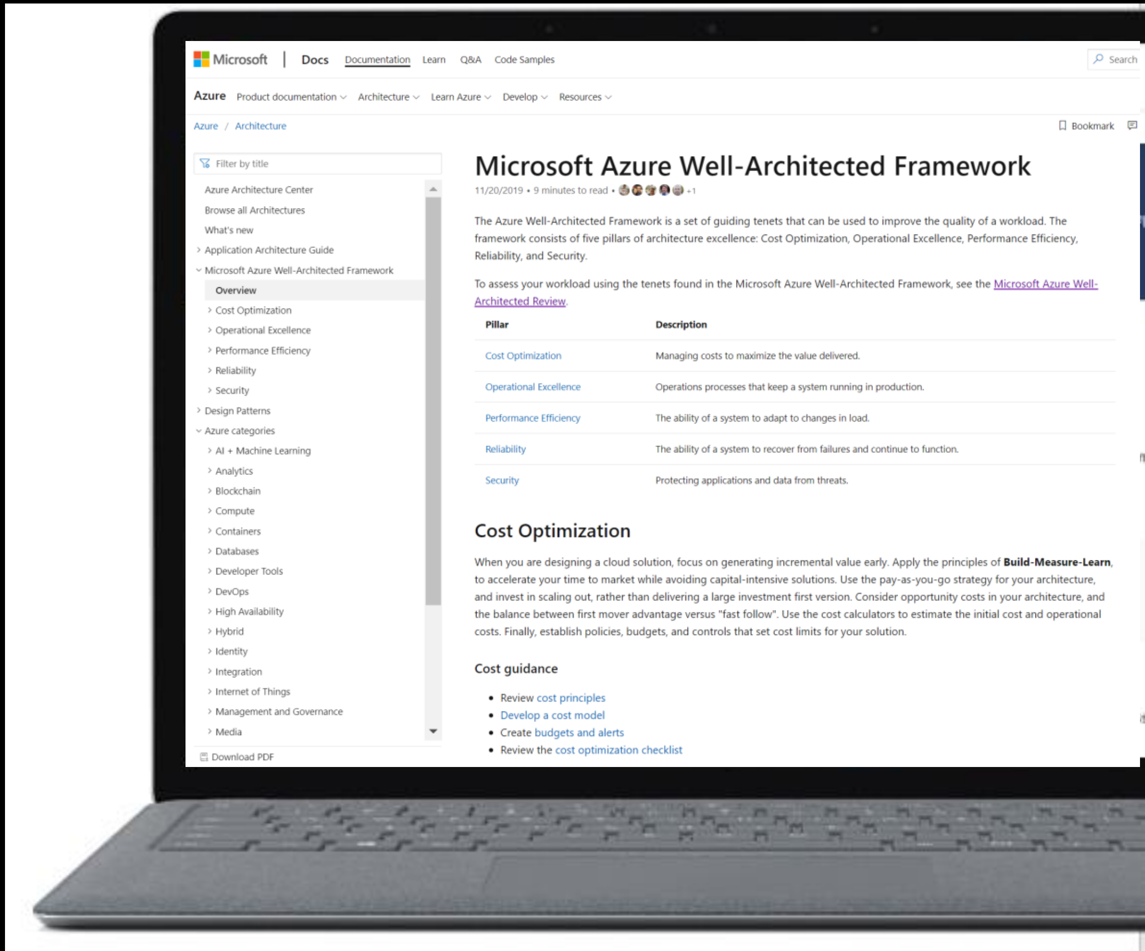
Designing for the cloud
- Principles of a well-designed application
- Web API design
- Building microservices on Azure
- Application design patterns
- Managing identity in multitenant apps

Optimizing your workload
- Guiding tenets for your architecture
- Examine your workload
- Performance tuning
- Performance antipatterns
- Securing your infrastructure

Documentation

Microsoft Azure Well-Architected Framework

https://aka.ms/wellarchitected/framework



Microsoft Azure Well-Architected Framework

11/20/2019 • 9 minutes to read • +1

The Azure Well-Architected Framework is a set of guiding tenets that can be used to improve the quality of a workload. The framework consists of five pillars of architecture excellence: Cost Optimization, Operational Excellence, Performance Efficiency, Reliability, and Security.

To assess your workload using the tenets found in the Microsoft Azure Well-Architected Framework, see the Microsoft Azure Well-Architected Review.

Pillar	Description
Cost Optimization	Managing costs to maximize the value delivered.
Operational Excellence	Operations processes that keep a system running in production.
Performance Efficiency	The ability of a system to adapt to changes in load.
Reliability	The ability of a system to recover from failures and continue to function.
Security	Protecting applications and data from threats.

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.

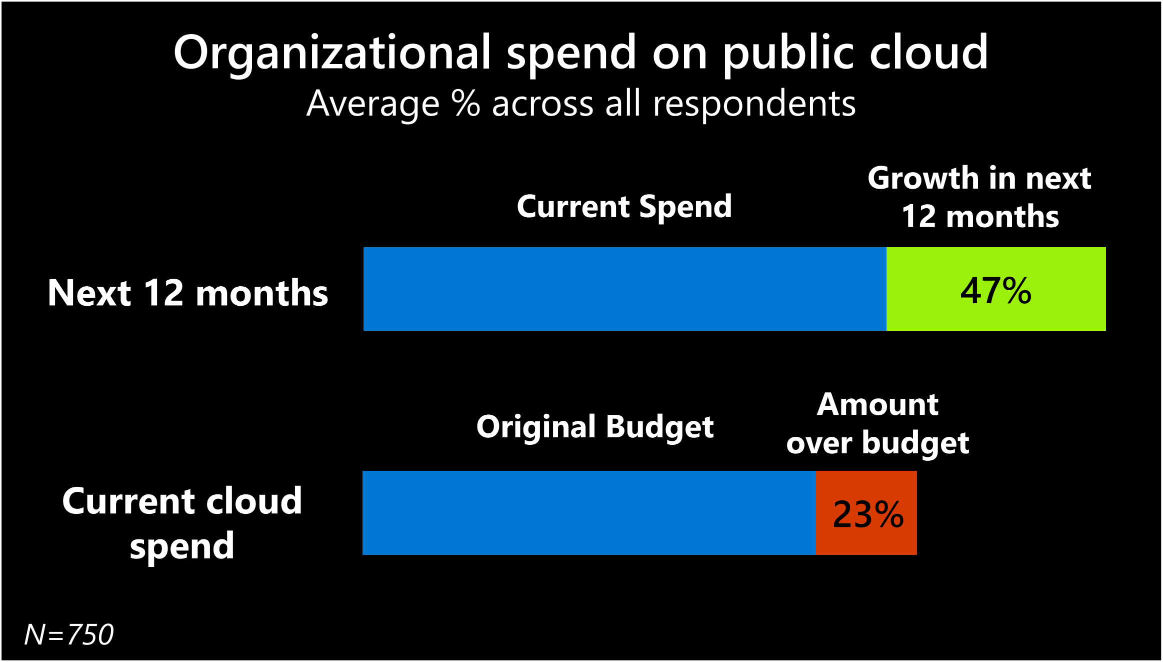
Cost guidance

- Review cost principles
- Develop a cost model
- Create budgets and alerts
- Review the cost optimization checklist

Well-architected solutions enable—cost optimization

It's more critical than ever for customers to get a handle on forecasting and cost optimization[1]

- Customers reported their public cloud spend was over budget by an average of 23 percent[1]
- Respondents expect their cloud spend to further increase by 47 percent in the next 12 months.



[1] *Flexera 2020 State of the Cloud Report*

Well-architected solutions enable—cost savings in security spend

In 2019, encryption, business continuity management, DevSecOps, and threat intelligence sharing mitigated cost[1]

- Encryption reduced breach costs by an average of $360,000.
- Business continuity management reduced the total cost of a data breach by an average of $280,000.

[1] The Cost of a Data Breach Report, IBM Security, 2019. Conducted by Ponemon Institute LLC

Well-architected solutions enable— cost savings with resiliency, high-availability, and security automation strategies

Companies with incident response teams with testing of IR plans —**saved over $1.2 million[1].**

Organizations without security automation experienced breach costs 95 percent higher

- Breach costs rose above 16 percent at organizations without automation deployed, going up from an average of $4.43 million in 2018 to $5.16 million in 2019.
- Breach costs decreased by 8 percent at organizations with fully deployed automation, from 2018 to 2019, from an average of $2.88 million in 2018 to $2.65 million in 2019.

[1] The Cost of a Data Breach Report, IBM Security, 2019. Conducted by Ponemon Institute LLC