# {Availability, Resilience, and Disaster Recovery}

# General Approach

# Define the problem

Common industry terms

- RTO – Recovery Time Objective
- RPO – Recovery Point Objective
- MTBF – Mean Time Between Failures
- MTTR – Mean Time to Recovery

A contract with your users and customers

- SLA – Service Level Agreement
- Perception and Reputation

# A resilient & available architecture

Understand the inter-dependencies between services inside and outside your solution – **and document them**.

- Could you list every component in the solution, along with it’s dependencies and sub-components?
- Could you draw a diagram of those components – i.e. your solution architecture on a whiteboard or piece of paper, marking key components ?
- Now could you annotate that diagram with performance and resilience data, i.e. Single points of failure and typical performance bottlenecks in day to day usage?
- Can you indicate where those components are geographically located on a map and if data is replicated elsewhere, indicate the mirrored region and the methodology of data movement / sync / mirroring etc?
- Can you show the entry point into your solution for a customer / end user (there may be multiple entry points) - i.e. which services are considered the 'front' of the stack - do the same on the geo map.
- Can you trace a path through the call stack of the application from that front end, stepping through each component used and region where data transits ?

# Azure Services

# Azure Service Availability

| Service | Monthly Uptime % | Specifics |
|---|---|---|
| Virtual Machines | 99.95% or 99.9% | Two or more instances, or single instance on premium storage |
| App Service | None or 99.95% | No SLA for apps deployed in the free or shared tiers |
| Service Fabric | Per underlying resource | Service Fabric is a free service with no specific SLA, based on underlying resource |
| Storage | 99.99%, 99.9% or 99% | SLA depends on geo, local or zone redundancy and storage tier |
| SQL Database | 99.99% | SLA is across the deployment options, excluding retired |
| Cosmos DB | 99.99% + Multiple | Latency, Throughput, Consistency, Availability |
| Azure Active Directory | None or 99.9% | No SLA on free tier, SLA on all other tiers. |

(User Minutes - Downtime) / User Minutes * 100 = monthly uptime %

99% = 403 minutes/month

99.99% = 4 minutes/month

As of Sept 2017

https://azure.microsoft.com/en-gb/support/legal/sla/summary/

# Azure Solution Resiliency

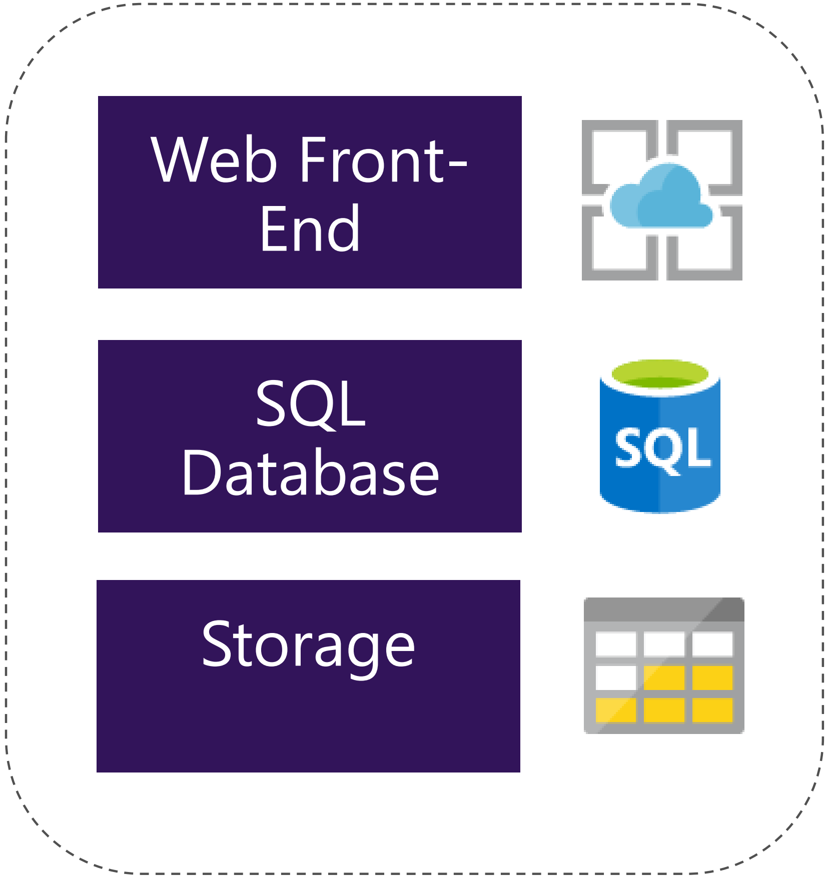
Your solution Azure SLA is the product of the service SLAs:

| | |
|---|---|
| Web Site hosted in App Service | = 99.95% |
| Azure SQL Database | = 99.99% |
| Storage | = 99.9% |
| **Solution SLA** | **= 99.84%** |

Your overall solution SLA needs to take account of *your* activity:

- Your application’s reliability and availability
- Deployment and DevOps activities
- Support and Recovery Operations
- ‘Unplanned’ activities
- …

# Using Azure – Thinking Resilience



From an architecture and design perspective, what are the options for availability and resilience?

How can issues effecting an Azure region be avoided and recovered from?

What are the cost implications of my availability and resilience choices?

Where can I get more detail?

# Using Azure – Storage

The application is using storage for image content and archiving data for future analysis.

## Image content

High user impact, high read volume, needs high availability

HOT Read Access Geo Redundant Storage (RAGRS – HOT)

Low cost to access data but higher cost to store due to HOT, automatically copied to another region which can be read by the application

## Archive Data

Not immediately useful but has longer term value that could be monetised

COOL Locally Redundant Storage (LRS – COOL) with manual backup

Very low cost storage due to COOL, may not be available occasionally but manually backed up on a monthly basis

https://azure.microsoft.com/en-gb/pricing/details/storage/blobs/

# Using Azure – SQL Database

The application is using Azure SQL database to store transaction data. The application is non functional without the database and information loss has high impact.

Performance analysis has selected the standard tier service.

The impact to users if the database is not present has selected active geo-replication to a single secondary region. Cost concerns mean lower secondary performance.

In the event of complete loss of online capability, standard tier service offers 35 days of backups with < 1 hour's lost data. We can restore from the replicated database.

https://azure.microsoft.com/en-gb/pricing/details/sql-database/

# Using Azure – App Service

The application is using ASP.NET to serve a web UI and a REST service layer. Both UI and service layer need to be running for the application to be functional.

Performance analysis has determined two instances of our web tier. This does not benefit our resiliency, there is a still a concern over user impact.

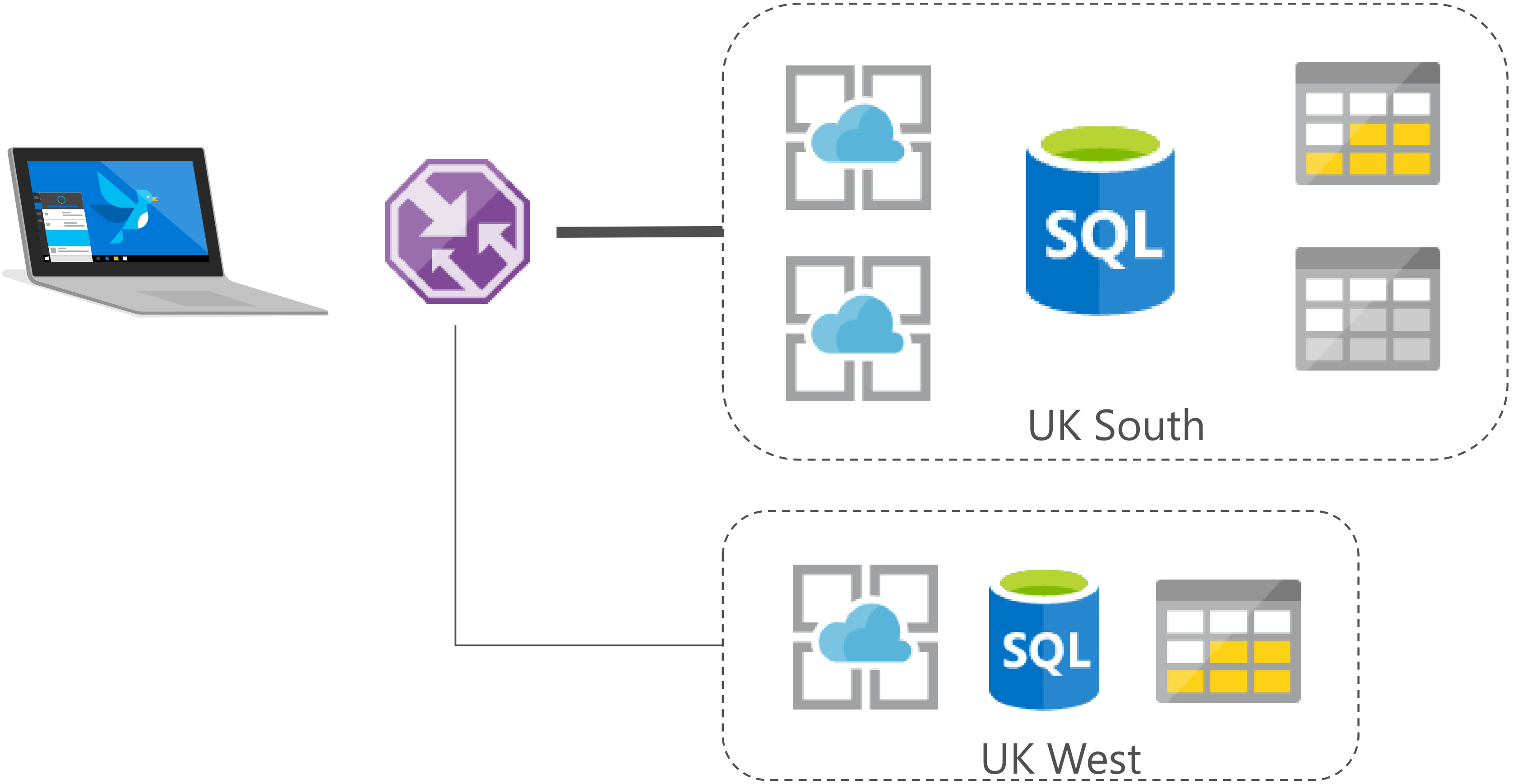
A replica of the web tier will be hosted in a second data centre, matching the location of our geo-replicated SQL database. We need to handle the second deployment as part of our DevOps. It will be scaled down to conserve cost.

Traffic manager will be used to allow users to switchover to the replica website. This will be configured to automatically switch over when failure is detected.

https://azure.microsoft.com/en-gb/pricing/details/app-service/

# Using Azure – Active/Passive

Our design choices have led to an Active/Passive deployment model for our application.



# Demo

## Kicking the application

# Using Azure – Other Models

IaaS – Azure Virtual Machine, Storage, Network SLA

Availability Sets, Managed Disks, Multi- site/region.

Software availability and resilience concerns.

PaaS-ier – Service Fabric, Containers, Functions. Pass-through SLA

Higher decomposition, reliance on PaaS orchestrators

Office 365 SAP SendGrid

SaaS – Direct SLA with the service provider or your on-premises investments. These are a dependency for your application and should be part of your planning.

Ignite

# Resilience

### SINGLE-REGION DEPLOYMENT IN AZURE

Some customers believe that Azure automatically deals with scalability and resiliency across regions.

*Customers should plan for an outage of services in a particular region and failover where necessary*

### LACK OF STRATEGY FOR RESILIENCE WITHIN SERVICES

Some Azure Services have functionality built in, to deal with availability.

*Customers should be aware of these services, e.g. Traffic Manager, or Service Bus  Paired Namespaces,* ***Storage GRS vs RA-GRS (and failover)***

### IGNORE THIRD PARTY DEPENDENCIES THAT COULD IMPACT YOUR SERVICE

Most solutions have included dependencies outside of Azure.

*Customers should ensure that a graceful degradation occurs, for components inside Azure and outside.*

### IGNORE SINGLE POINTS OF FAILURE

Some solutions we have seen have single points of failure in their solution. If this goes down, your application will be down.

*Customers should run all tiers of their application in a resilient manner if the SLA requires it.*

# Information and Processes

# Plan for recovery

The application has been architected for resilience but your people and processes need to be planned.

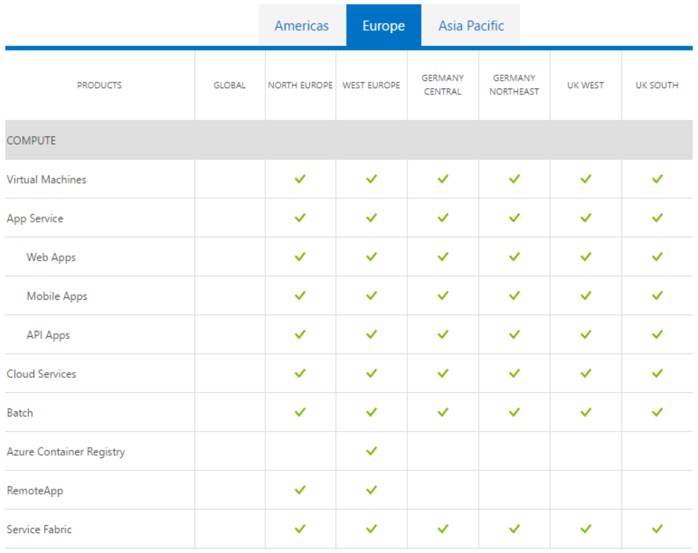
**The major incident response plan (MIRP)**

Who is monitoring for failure?
Who needs to be alerted when failure occurs?
Does engineering have the correct access/permissions?
How will troubleshooting information be captured?
How would you redeploy the application from scratch?
How do you contact your suppliers (including Microsoft)?
Can you categorise and prioritise the severity of the incident?

# Monitor your applications, use the data

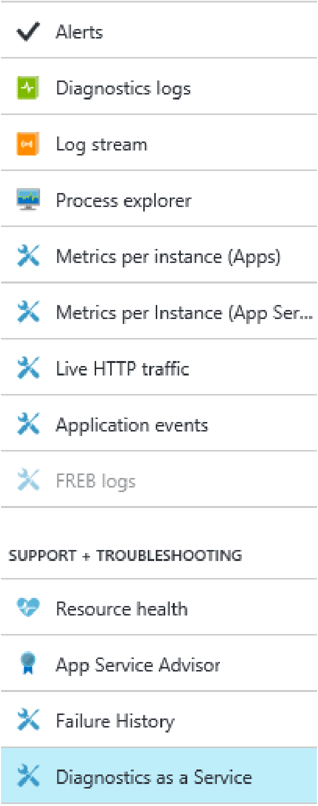
Discover the health of Azure and your application.  Use that information to respond and categorise.

## Azure status dashboard and Health Blade

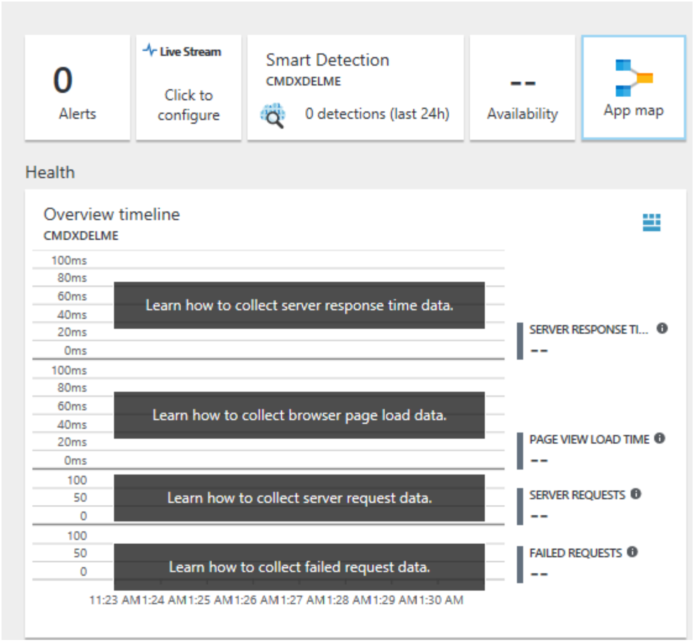


| PRODUCTS | GLOBAL | NORTH EUROPE | WEST EUROPE | GERMANY CENTRAL | GERMANY NORTHEAST | UK WEST | UK SOUTH |
|---|---|---|---|---|---|---|---|
| COMPUTE | | | | | | | |
| Virtual Machines | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| App Service | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Web Apps | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Mobile Apps | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| API Apps | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Cloud Services | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Batch | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Azure Container Registry | | | ✓ | | | | |
| RemoteApp | | ✓ | ✓ | | | | |
| Service Fabric | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

## Diagnostics, Logs and Alerts with your service



## Application Insights to monitor and diagnose



Learn how to collect server response time data.

Learn how to collect browser page load data.

Learn how to collect server request data.

Learn how to collect failed request data.

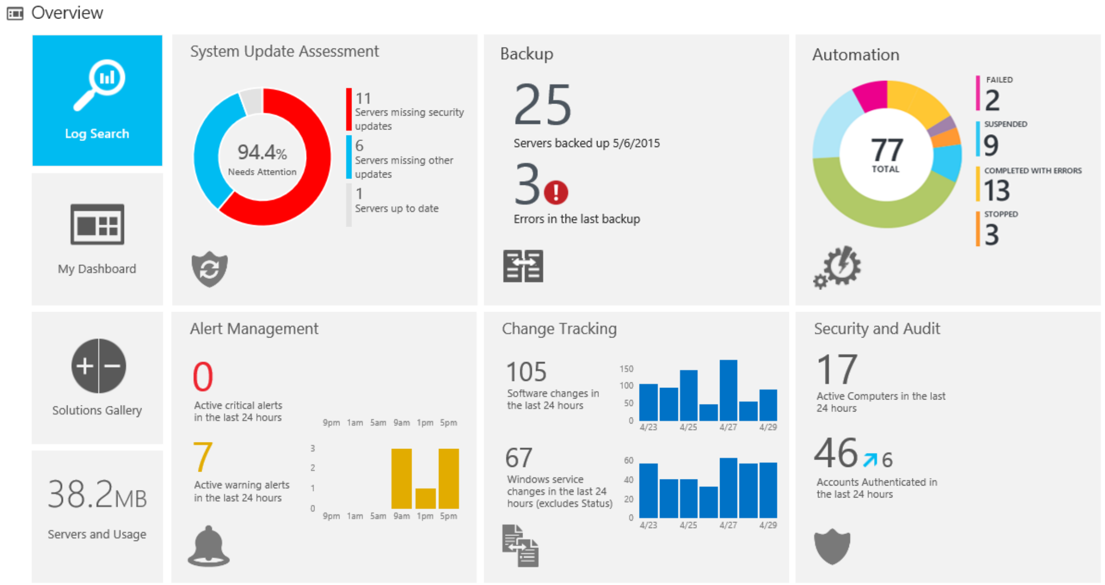
https://azure.microsoft.com/en-gb/status/

Ignite

# Operational Monitoring

## At scale there are tools to build operational processes and insight into your BAU running of applications

### Microsoft Operations Management Suite



Overview

Log Search

My Dashboard

Solutions Gallery

38.2MB
Servers and Usage

System Update Assessment

94.4%
Needs Attention

11
Servers missing security updates

6
Servers missing other updates

1
Servers up to date

Backup

25
Servers backed up 5/6/2015

3
Errors in the last backup

Automation

77
TOTAL

FAILED
2

SUSPENDED
9

COMPLETED WITH ERRORS
13

STOPPED
3

Alert Management

0
Active critical alerts in the last 24 hours

7
Active warning alerts in the last 24 hours

Change Tracking

105
Software changes in the last 24 hours

67
Windows service changes in the last 24 hours (excludes Status)

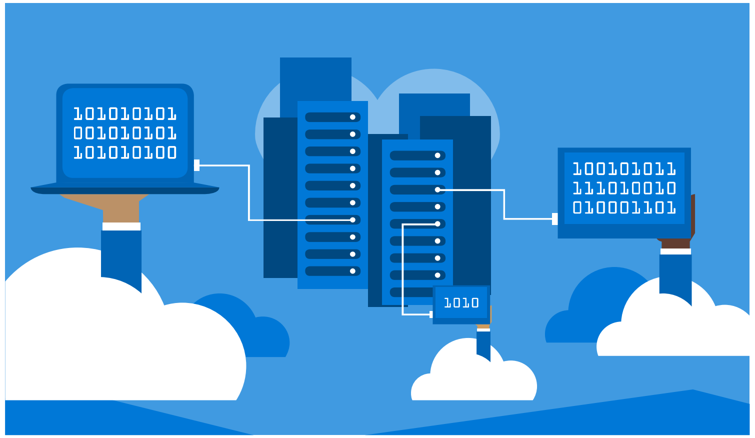
Security and Audit

17
Active Computers in the last 24 hours

46 ↗ 6
Accounts Authenticated in the last 24 hours

A tool for end to end IT operations and monitoring of multiple complex application deployments.

### Third Party Partner Tools and Services



Variety in functionality and price point for your needs and objectives

# Demo

## Monitoring the application

# Availability and Disaster Recovery

### Ignore a major incident response plan

In a number of assessments, customers did not have a Major Incident response plan in place.

*Customers should ensure that they have a MIRP in place, clearly defining responsibilities across the solution, escalation protocol and any other necessary processes to follow in a disastrous scenario.*

### Ignore the need for a data reconciliation strategy

Many solutions are based around the concept of Eventual Consistency and / or multi-region deployment.

*When faced with region failover, how do you know that critical data is present and valid in the new region, what was lost from the primary region and may now never arrive in the secondary ?*

### Fail to test your DR / HA strategy

A number of customers had either an automated Disaster Recovery mechanism or well-documented approach. However, this approach had not yet been tested.

*Customers should test their disaster recovery semi-regularly, to ensure that the process is still relevant and that all parties are aware of the required steps.*

# Resources

# Resources

**Designing resilient applications for Azure**

https://docs.microsoft.com/en-us/azure/architecture/resiliency/

**Overview of business continuity with Azure SQL Database**

https://docs.microsoft.com/en-us/azure/sql-database/sql-database-business-continuity

**Availability and Resiliency Checklists**

https://docs.microsoft.com/en-us/azure/architecture/checklist/availability

**Azure Monitoring Overview**

https://docs.microsoft.com/en-us/azure/monitoring-and-diagnostics/monitoring-overview

**Azure Service Level Agreements**

https://azure.microsoft.com/en-gb/support/legal/sla/

**Azure Support Plans**

https://azure.microsoft.com/en-gb/support/plans/