Session 18
Monitoring (native)

Anton Boyko
Microsoft Regional Director
Microsoft Azure MVP
me@boykoant.pro

Housekeeping

- Please keep yourself muted unless you are participating in the conversation, so we can have a more clear recording.
- If you have questions – don’t hesitate and ask.

True story

How do you know if your app is working?

- If the app is not working, our users will call IT Support and they in turn will call us.
- We measure CPU load. If it is above X%, we'd start investigation because there may be an issue.
- We stream all the logs to ELK stack.
- We do web pings with Pingdom.
- We monitor it with Prometheus and display metrics with Grafana.

Create common ground

Monitoring

Monitoring - observe and check the progress or quality of something over time; keep under systematic review.

Why to monitor:

- Problem detection
- Troubleshooting
- Reporting and improvement

Observability

Observability - (quality attribute) a measure of how well the internal states of a system can be inferred from knowledge of its external outputs.

There can be general measurements like CPU load, RAM load, etc. But there are also can be the custom measurements. A good example of custom measurement can be the amount of payment transactions that were handled by your application.

White box vs black box

- White box – monitor system's internals.
- Black box – "emulate" the behaviour of end users.

Types of measurements

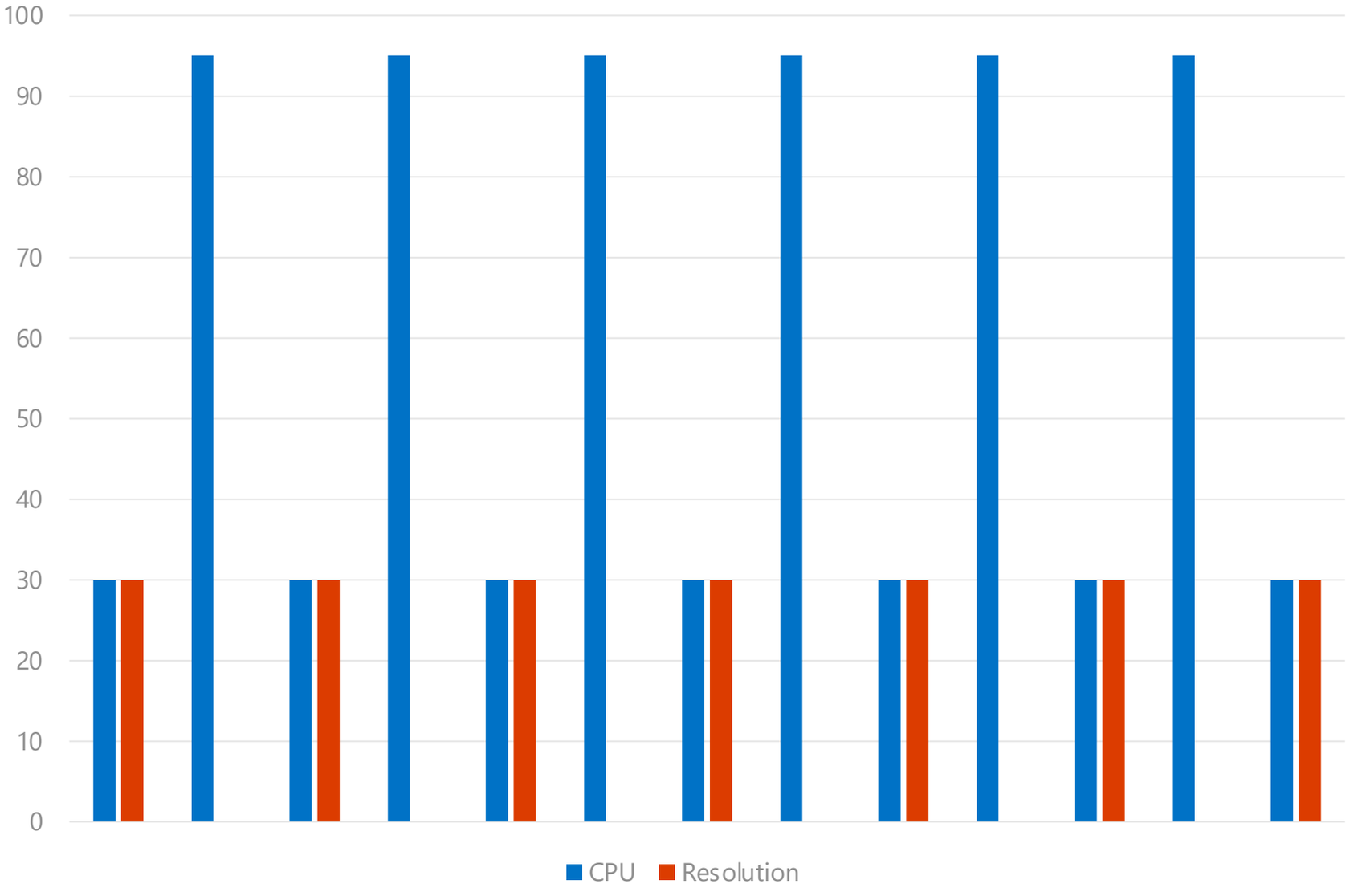
- Metrics
- Events (emits when a predefined set of conditions is met)
- Diagnostics (contains deep data about the system state, but often requires a detailed analysis to get some useful information)

Metrics precision:

- What is its value?
- How is it sampled?
- Is it exact or aggregated?
- What is its frequency?
- What time is it from?
- What does it mean?

Resolution

The wider the period, the less extra load you will put on your system that is under monitoring, but the bigger the opportunity that you will miss something if it occurs and disappears between the start and stop marks of your monitoring period.



Aggregation

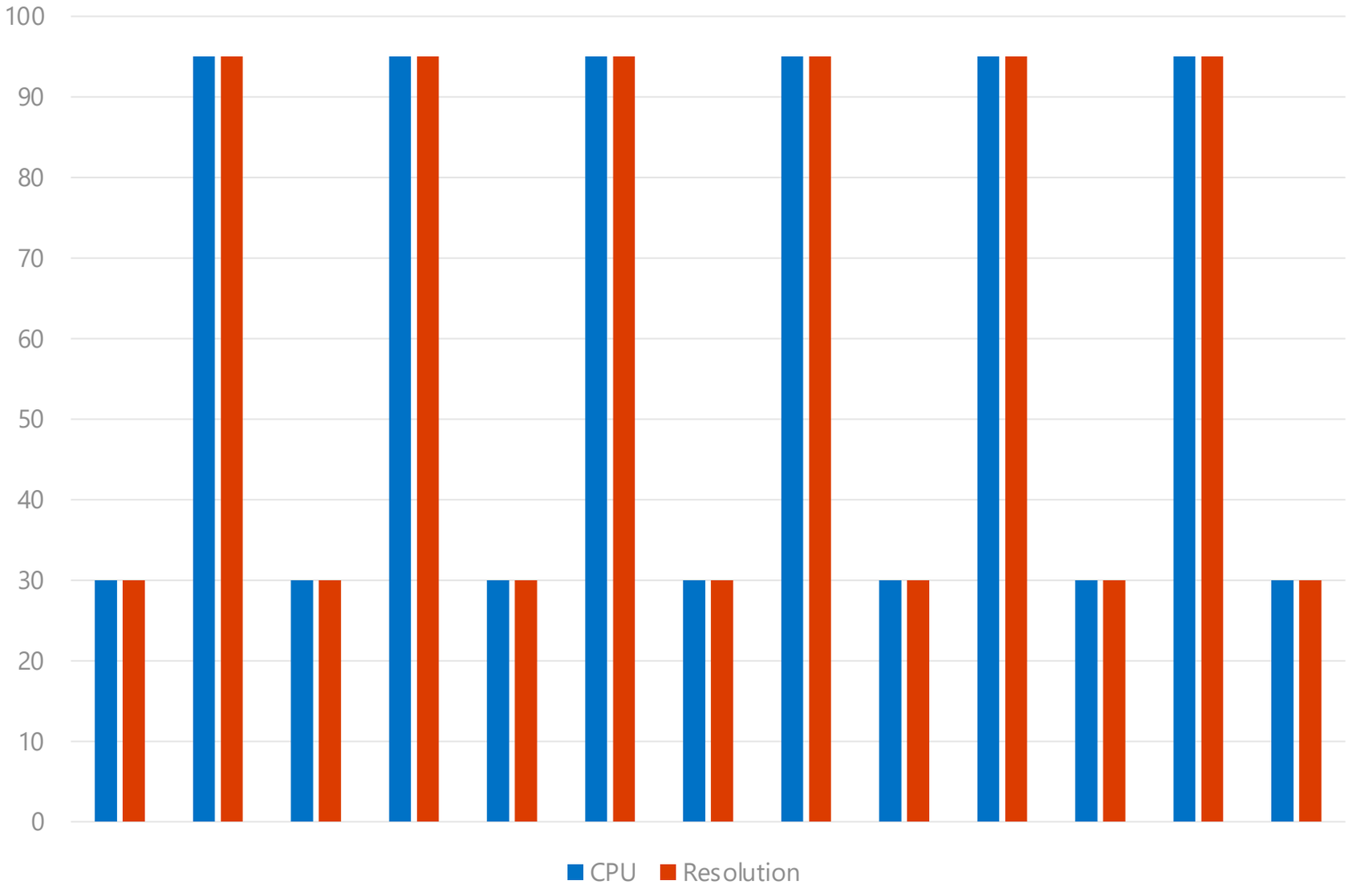
AVG = 60
P50 = 30
P90 = 95

AVG < P90

Load goes up and down.

AVG > P90

Load is usually small, but sometimes a huge spike can occur.



USE model

USE model

Utilization, Saturation, Errors.

This model is suitable mostly for infrastructure. It was originally created by a performance expert to quickly analyze the performance of a system and easily find the most crucial bottlenecks. In any case - it can be adapted from monitoring performance to monitoring infrastructure in general.

Terminology

- Resource: all physical server functional components (CPUs, disks, busses, ...).
- Utilization: the average time that the resource was busy servicing work.
- Saturation: the degree to which the resource has extra work which it can't service, often queued.
- Errors: the count of error events.

aboyko-demo-03 | Metrics

Virtual machine

Search (Cmd+/)

Auto-shutdown
Backup
Disaster recovery
Guest + host updates
Inventory
Change tracking
Configuration management (Preview)
Policies
Run command

Monitoring

Insights
Alerts
Metrics
Diagnostic settings
Logs
Connection monitor (classic)
Workbooks

Automation

Tasks (preview)
Export template

Support + troubleshooting

Resource health
Boot diagnostics

New chart | Refresh | Share | Feedback | UTC Time: 9/8 12:00 PM - 9/8 9:45 PM (1 minu...

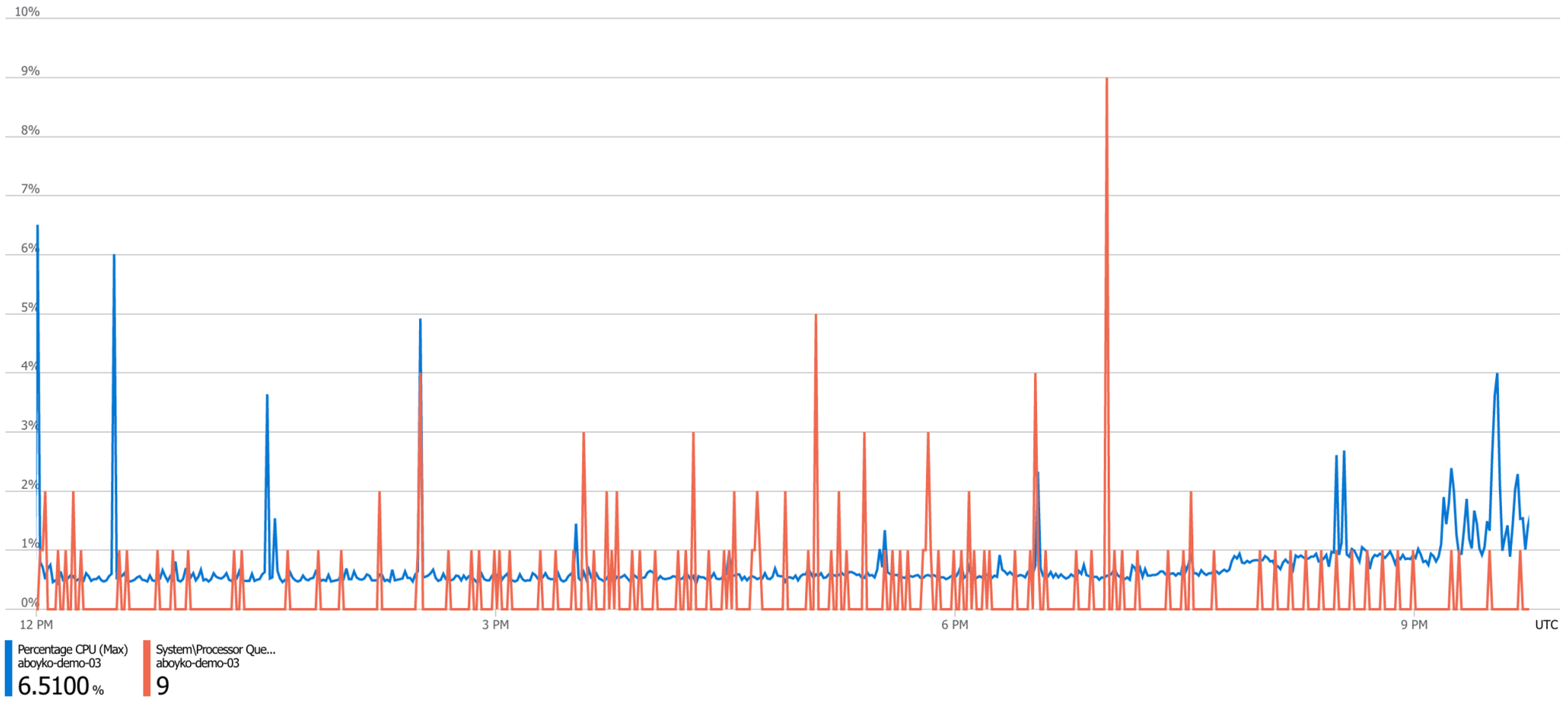
Max Percentage CPU and Max System\Processor Queue Length for aboyko-demo-03

Add metric | Add filter | Apply splitting | Line chart | Drill into Logs | New alert rule | Pin to dashboard

aboyko-demo-03, **Percentage CPU,** Max

aboyko-demo-03, **System\Processor Que...** Max

Undo Zoom

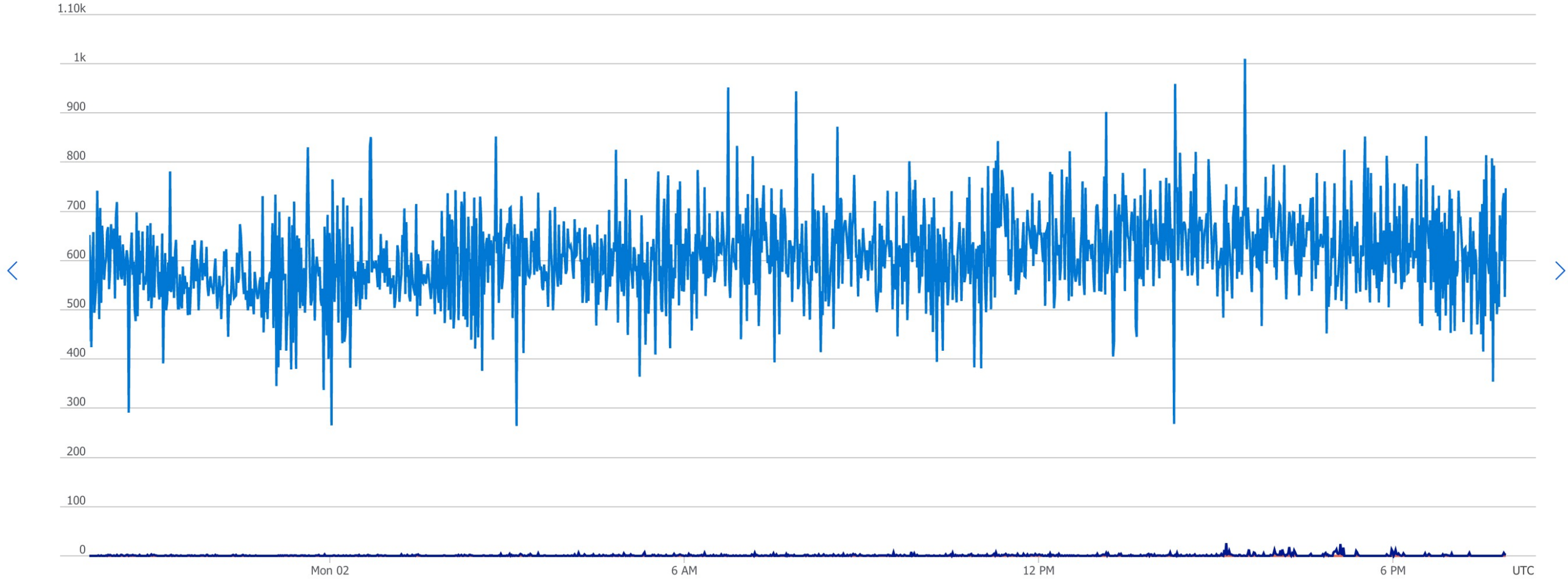


New chart | Refresh | Share | Feedback | UTC Time: Last 24 hours (1 minute)

Sum Requests, Sum Requests In Application Queue, and Sum Http Server Errors for

Add metric | Add filter | Apply splitting | Line chart | Drill into Logs | New alert rule | Pin to dashboard | ...

Requests, Sum | **Requests In Applicatio...** Sum | **Http Server Errors,** Sum



Requests (Sum)
prd-web-fe-01
875.96 k

Requests In Applicat...
prd-web-fe-01
0

Http Server Errors (...
prd-web-fe-01
1.29 k

Small utilization and big saturation

Condition

- Application is hosted on top of Standard_D2_v4 (2 CPU, 8Gb RAM) VM.
- CPU utilization is 50% for the last 10 minutes.
- CPU saturation is 5 for the last 10 minutes.

Interpretation

- Application is executing a CPU-heavy long-running single-thread job.
- Application code is bound to a single CPU core, probably because it is written very poorly.

DITCHING

PREPARATION

1. Radio - MAYDAY (route freq), IdentifyPosition, Heading, Altitude, IAS.
2. Locate and Head for Nearest Ship.
3. Notify Crew & Cabin to Perform Duties.
4. Assign Life Raft Locations if Advisable.
5. Turbos & Bleeds - OFF , 10,000 Ft.
6. Depressurize Cabin - Reset Controls to Pressurize to insure Outflow Valves are Closed.
7. Close Thrust Valves.
8. Secure Loose Equipment.
9. No Smoking & Seat Belt - ON
10. Horn & Alt Warn CB P5;(331 P6) - PULL
11. Set Altimeters to Local Pressure, Cross check Radio Altimeters 1000 Ft & Below.
12. Fuel Panel - TANK TO ENGINE
13. Dump Fuel To Standpipes - Close Valves
14. Retract Dump Chutes
15. Crew Life Vests; Shoulder Harness; & Seat Belt - SECURE

EMERGENCY LAND LANDING

PREPARATION

1. Radio - MAYDAY (route freq), Identify Position Heading, Altitude, IAS.
2. Notify Crew & Cabin to Perform Duties.
3. Turbos & Bleeds - OFF, 10,000 Ft.
4. Exits - CLOSED
5. Secure Loose Equipment.
6. No Smoking & Seat Belt - ON
7. Set Altimeters to Local Pressure.
8. Dump Fuel if Advisable.
9. Retract Dump Chutes.
10. Seat Belts & Shoulder Harness — SECURE.

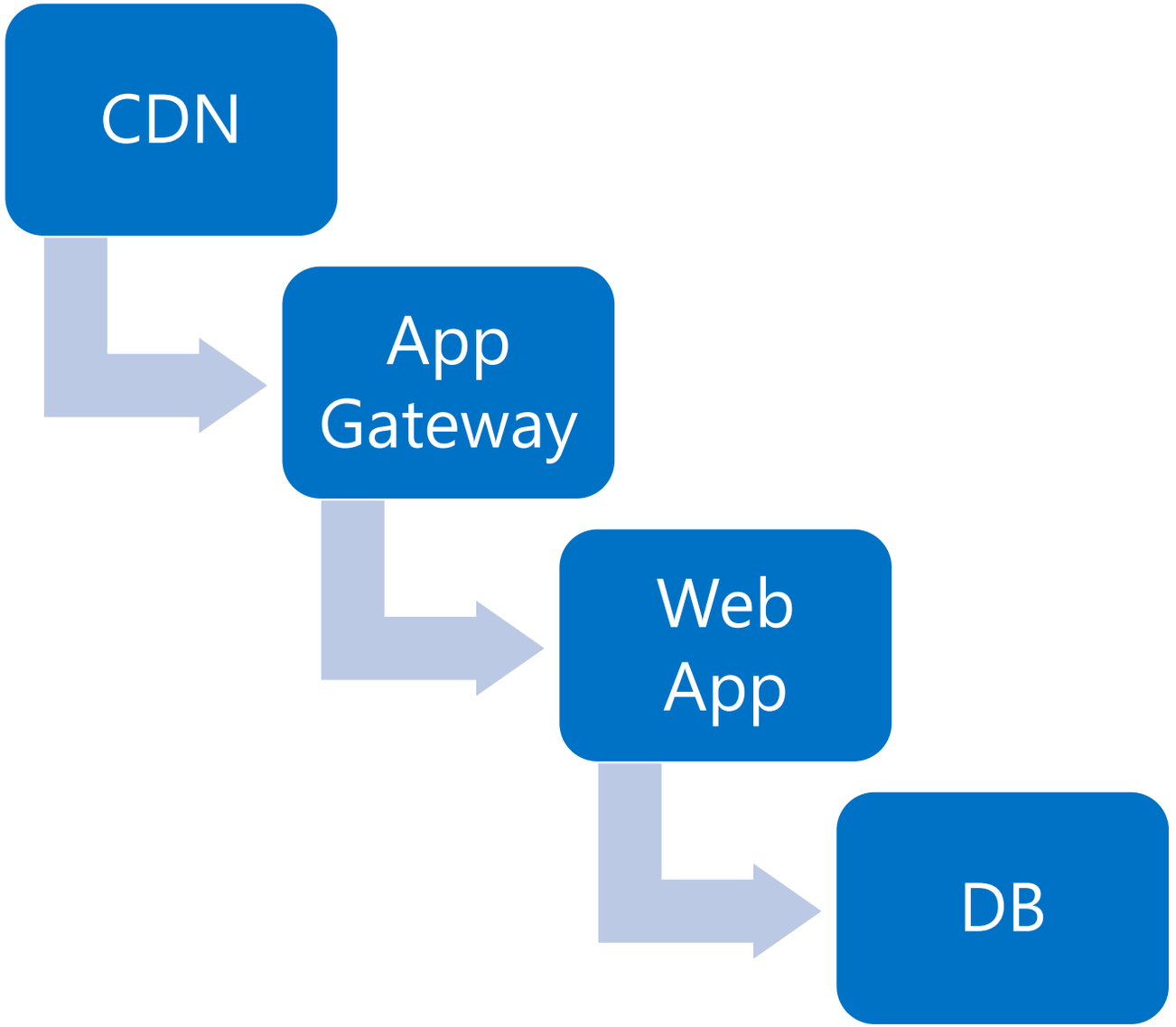
FINAL

1. Gear & Flaps - AS DESIRED.
2. Warn Crew & Cabin at 500 Ft.
3. Emergency Exit Lights - ON
4. Night - Main Cabin Lights - OFF
5. Command at 50 Ft. - BRACE FOR IMPACT.

Checklist example

Automated tool reports that our website homepage returns 404.

- Check homapege manually or with other tool.
- Check direct URL of Web App (bypass CDN and App Gateway).
- Check direct URL of App Gateway (bypass CDN).
- Purge CDN cache.



When you have eliminated the impossible, whatever remains, however improbable, must be the truth

Sherlock Holmes
The Sign of the Four (1890)
Chapter 6, paragraph 111
by Sir Arthur Conan Doyle

RED model

RED model

Requests, Errors, Duration.

RED model was created as a simplified version of Google's Four Golden Signals model. The main goal is to monitor a microservices based systems, especially when several microservices can be hosted on top of a single physical/virtual hardware unit.

Requests

Unscoped

Requests (count)	10 000
Duration (avg)	700 ms
Errors (percentage)	0.1 % (10)

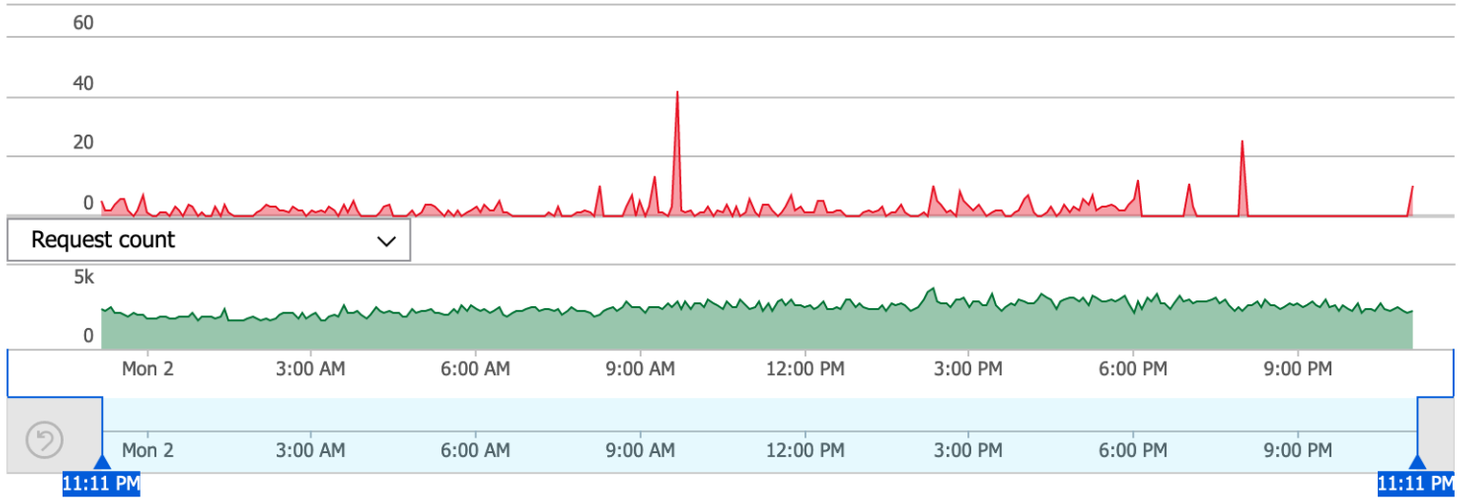
Scoped

Requests (count)	10
Duration (avg)	60 000 ms
Errors (percentage)	100% (10)

Server | Browser | Local Time: Last 24 hours | Roles = All

Operations | Dependencies | Exceptions | Roles

Failed request count



Select operation

Search to filter items...

OPERATION NAME	COUNT (FAILED)	COUNT	PIN
Overall	**535**	**688.86k**	
POST /identity/externallogin	305	1.64k	
POST /identity/externallogincallback	83	413	
POST /coveo/rest/v2	41	6.49k	
GET /	18	34.43k	
GET /sitecore/api/layout/render/jss	15	137.73k	
GET /orthopedic-trauma/adult-trauma/tibial-shaft/multifragmentary-fracture-fragmentary...	1	9.26k	
GET /spine/trauma/subaxial-cervical/basic-technique/cervical-pedicle-screw-insertion	1	50	
GET /-/media/surgery/34/34_p080_i160.ashx	1	36	

Overall

Top 3 response codes

	COUNT	FILTER...
500	438	
429	41	
404	29	

Top 3 exception types

	COUNT	FILTER...
Error [ERR_HTTP_...	48	
HttpException	30	

Top 3 failed dependencies

	COUNT	FILTER...
Http	971	
Http (tracked com...	50	

Drill into...

535 Samples

New Query 1*

Feedback | Queries | Query explorer

c1-eu-prd-01-oms... | Select scope | Run | Time range : **Set in query** | Save | Share | New alert rule | Export | Pin to dashboard | Format query

```
AppServiceHTTPLogs
| project Time = TimeTaken * 1ms, CsHost, CsMethod, CsUriStem, CsUriQuery, ScBytes, ScStatus
| where
Time > 0ms
and not(CsHost contains ".scm.")
and not(CsHost contains "nginx")
and not(CsUriStem contains "sitemap.xml")
and ScStatus >= 200
and ScStatus < 400
| summarize
TimeMax = max(Time),
```

Schema and Filter

Results | Chart | Columns | Display time (UTC+00:00) | Group columns

Completed. Showing results from the custom time range. 00:21.4 | 200 records

CsUriStem	CsHost	TimeMax	TimeAvg	TimeMin	Time90
/	0.azurewebsites.net	00:02:02.39900...	00:00:47.23366...	00:00:00.0150000	00:02:0

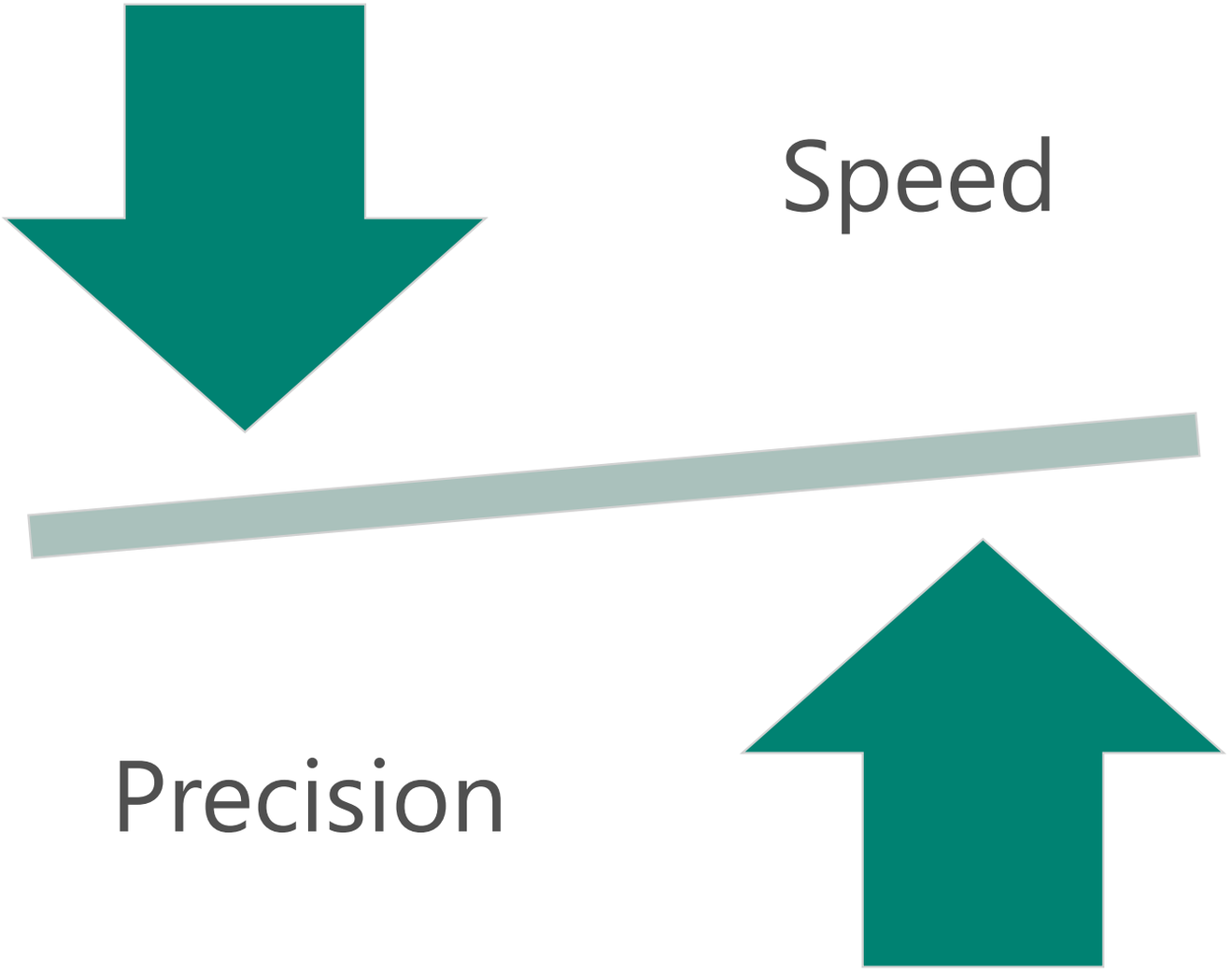
CsUriStem	/
CsHost	0.azurewebsites.net
TimeMax	00:02:02.3990000
TimeAvg	00:00:47.2336666
TimeMin	00:00:00.0150000
Time90	00:02:02.3990000
Time95	00:02:02.3990000
Time99	00:02:02.3990000
Bytes	396
count_	3

Page 1 of 1 | 200 items per page | 1 - 200 of 200 items

DEMO

Measurements

Balance



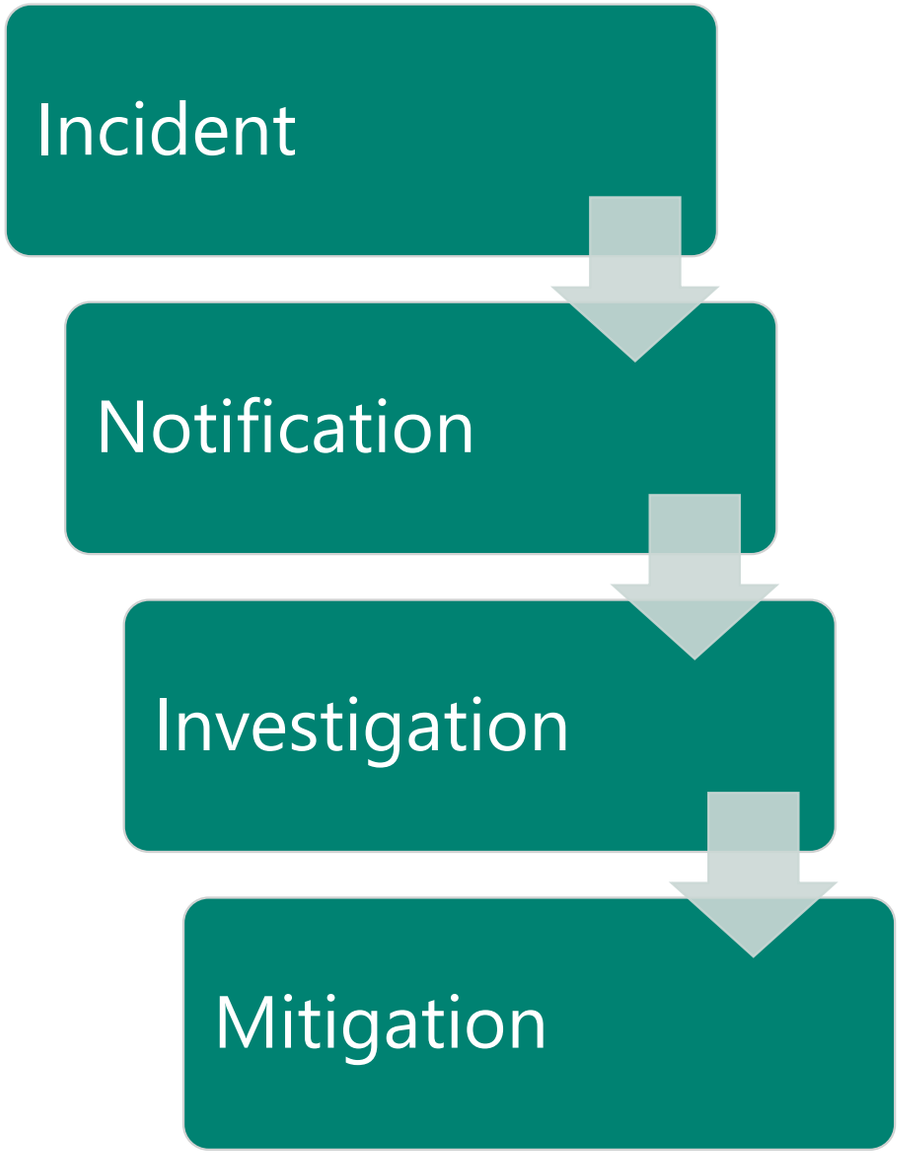
Custom healthcheck API

- Application runtime (simple echo endpoint)
- Internal dependencies (database, cache, storage)
- External dependencies (3rd party components)

MTTD

Mean Time To Detect

How long have the problem existed before you became aware of it?



MTTR

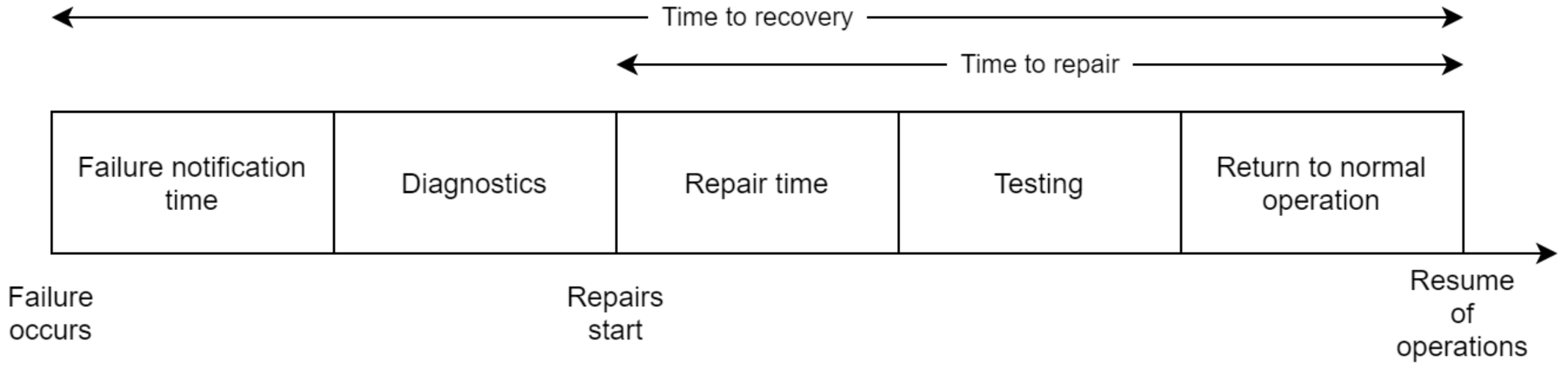
Mean Time To Repair

Amount of time required to repair a system and restore it to full functionality.

MTTD and MTTR

You can’t improve what you don’t measure

If monitoring tools and processes work as intended, it should help you to keep your MTTD and MTTR low



Questions?

Homework

Do (1)

- Create infrastructure templates using ARM, Bicep and Terraform.
- Create templates for App Insights web availability tests.

Do (2)

Create template for Azure Dashboard:

- ❑ App Service Plan
  - CPU (avg, max)
  - RAM (avg, max)
  - Disk queue (avg, max)
  - HTTP queue (avg, max)
- ❑ App Service Plan name and Resource Group name must be mandatory parameters

Do (3)

Create template for Azure Dashboard:

- App Service
  - Requests (sum)
  - Requests duration (avg, max)
  - Requests in queue (avg, max)
  - CPU time (sum)
  - Memory (avg, max)
  - HTTP statuses (sum)
    - 100
    - 200
    - 300
    - 400
    - 500
- App Service Plan name, App Service name and Resource Group name must be mandatory parameters

Do (4)

Create template for Azure Dashboard:

- Application Gateway
  - Total requests (sum)
  - Failed requests (sum)
  - Healthy / unhealthy hosts count (avg)
  - Throughput (avg)
  - Application gateway total time (avg)
  - Backend connect time (avg)
  - Backend first byte (avg)
  - Backend last byte (avg)
- Application Gateway name and Resource Group name must be mandatory parameters

Deadline

Recommended – by the end of the day 16.09.2022

Maximum – by the end of the day 23.09.2022