

HOW DO YOU HANDLE THE LOAD? SCALING SOLUTIONS ON AZURE

Gavin Barron

Principal Solution Architect

Intergen



EUROPEAN COLLABORATION SUMMIT



EUROPEAN CLOUD SUMMIT

29TH NOVEMBER – 1ST DECEMBER 2021



AGENDA

Scale basics

Monitoring

Architecting for scale

Wrap up



Gavin Barron

 @gavinbarron

 gavinbarron

Principal Solution Architect

- Team Lead for Microsoft Digital Events
- 8x MVP
- Building on Azure since 2011
- <https://gavinb.net>





Validate the business case!





Scale Basics - Hosting Tiers

Free

Up to 10 apps in a shared environment

Shared

Up to 100 Dev/test apps with SSL, domain names etc. best for non-production or for low traffic sites.

Basic

Apps with lower traffic requirements. Host unlimited websites/domains and includes built-in network load balancing

support.

Standard

Designed for production apps. Host unlimited websites/domains and includes built-in network load balancing, auto-scale, and

backup support.

Premium

Provide enhanced performance for production apps with faster processors, SSD storage, and double memory-to-core ratio

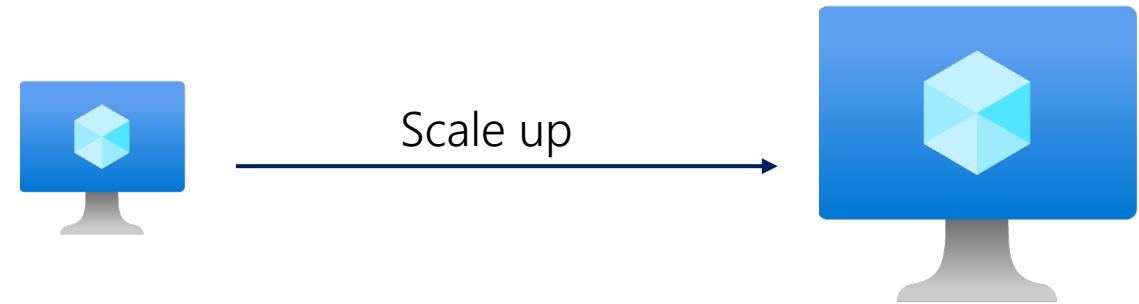
Isolated

compared to Standard
High-Performance, Security and Isolation



Scale Basics – Scale up

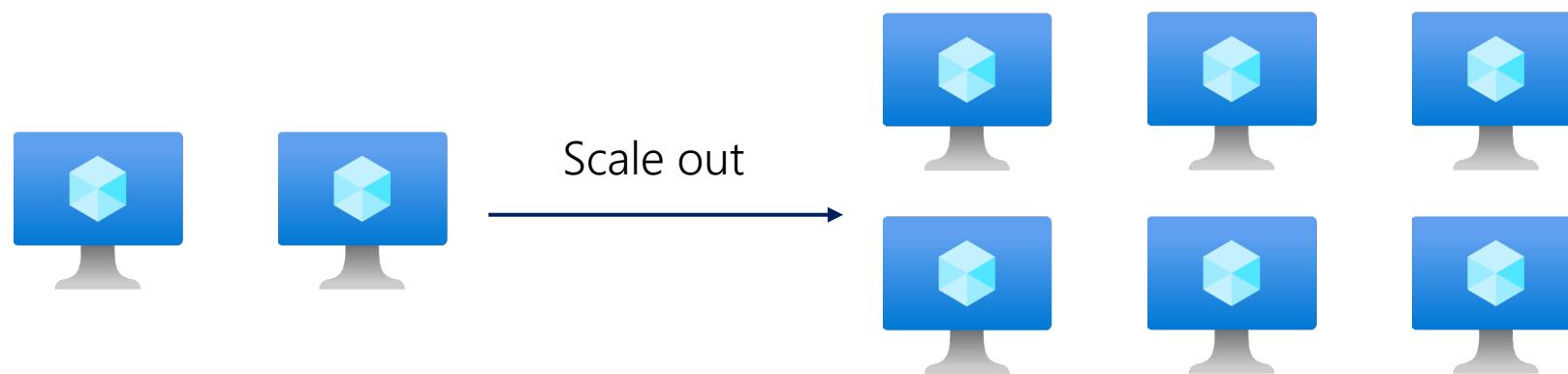
- Give your apps more RAM and CPU to work with
- Change SKU levels to add/remove capabilities
- P1v2
 - 1x core, 3.5 GB RAM
- P2v3
 - 2x core, 7 GB RAM
- P3v2
 - 4x core, 14 GB RAM





Scale Basics – Scale out

- Add additional instances to handle load
- Automatically load balanced
- Manual or Auto scale options





Scale Basics - Auto scale

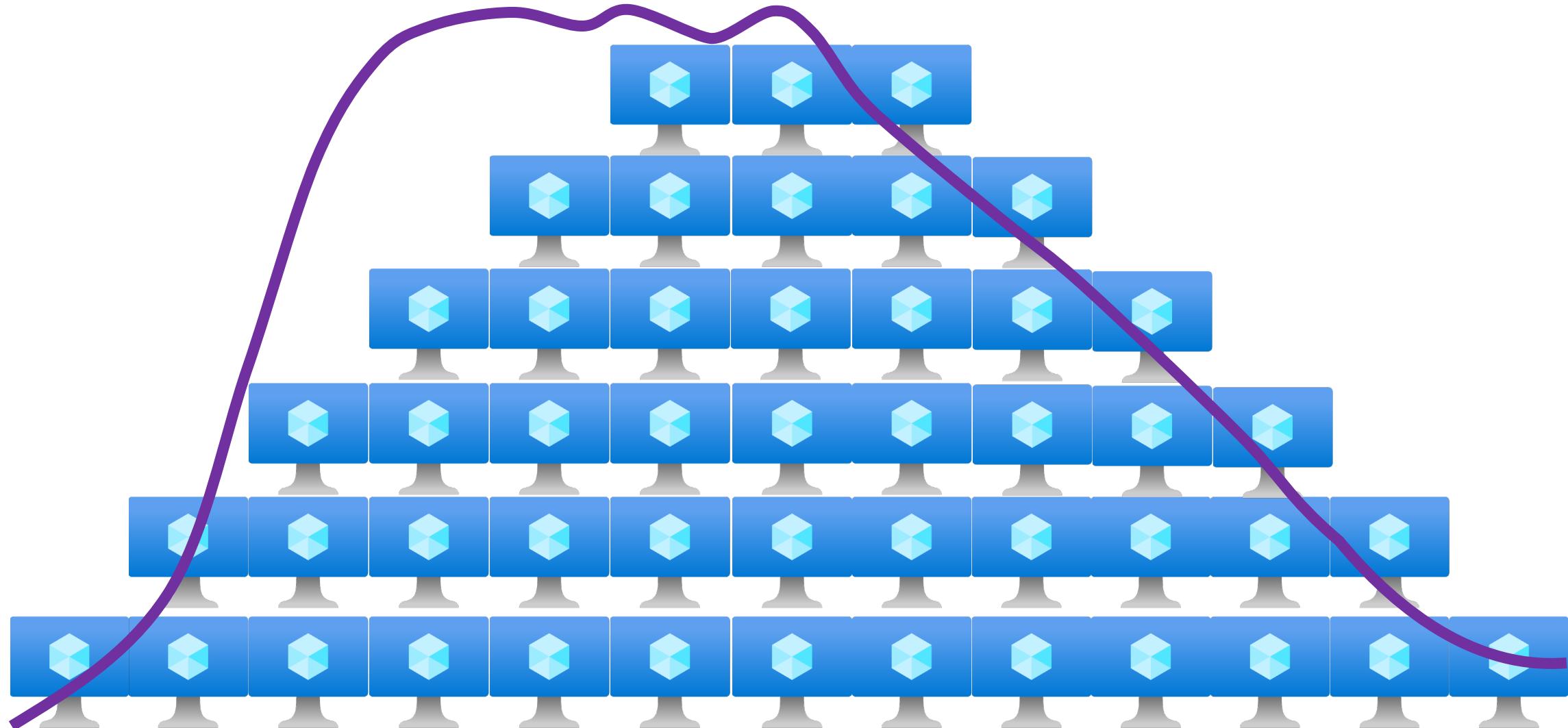




SCALE CONFIGURATION



Scale Basics - Auto fail





Scale Basics - Auto scale recap

Automatically add or remove server instances

Triggers based on metrics per time interval

Requires careful tuning

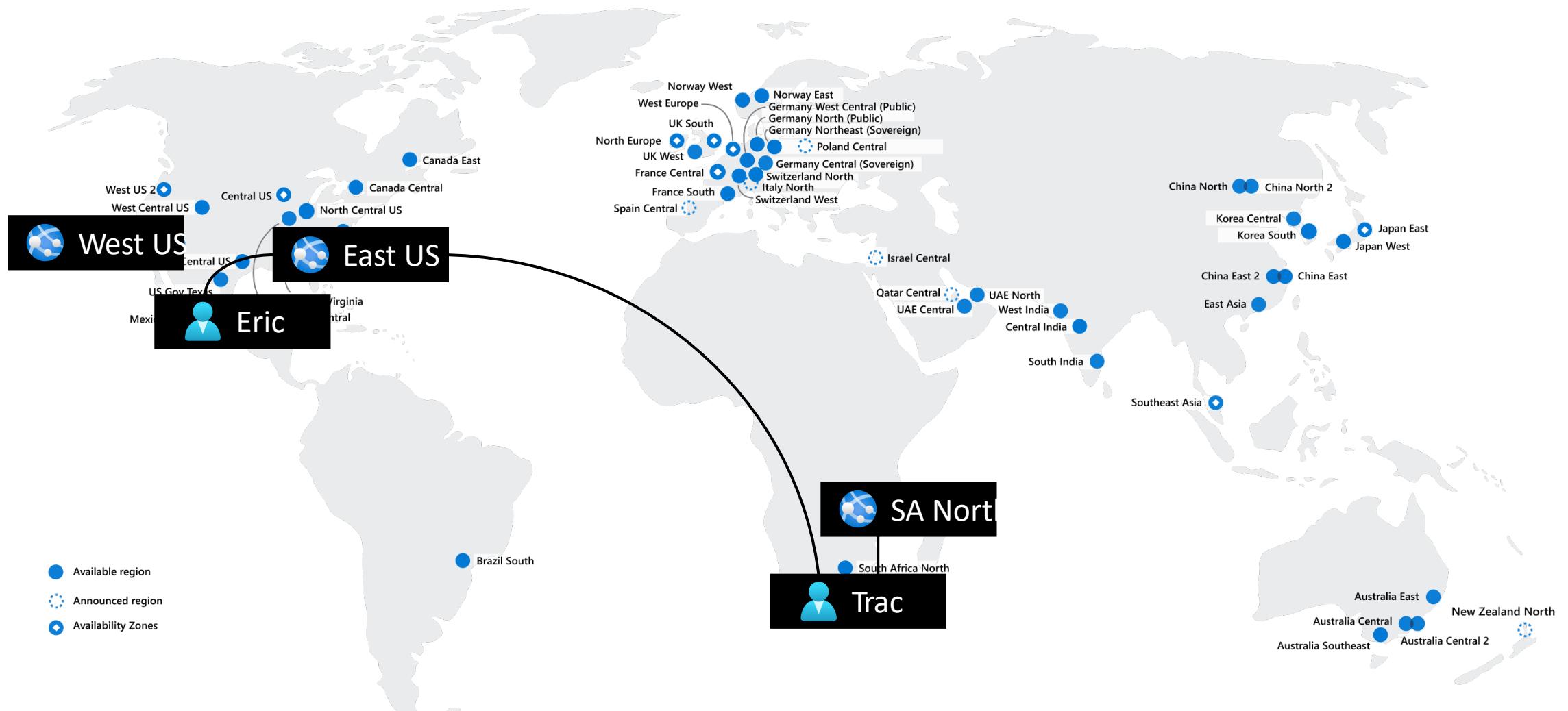
Start up time and state management

~~considerations~~
NOT a magic cure-all

Useful to handle normal traffic fluctuations



Scale Basics – Location





Scale Basics – Spanning Data Centers

Traffic Manager

- Cross region redirection & availability
- Any protocol
- DNS Based routing
 - On premises routing
- SSL termination at underlying infrastructure
- BYO SSL Certs only
- No additional security services
- No caching

Front Door

- Cross region redirection & availability
- http(s) only
- Request based routing
 - Routing rules based on path segments
- SSL offloading
- BYO or have SSL Cert provided
- WAF included
- Static content caching available



TRAFFIC MANAGER & FRONT DOOR CONFIGURATION

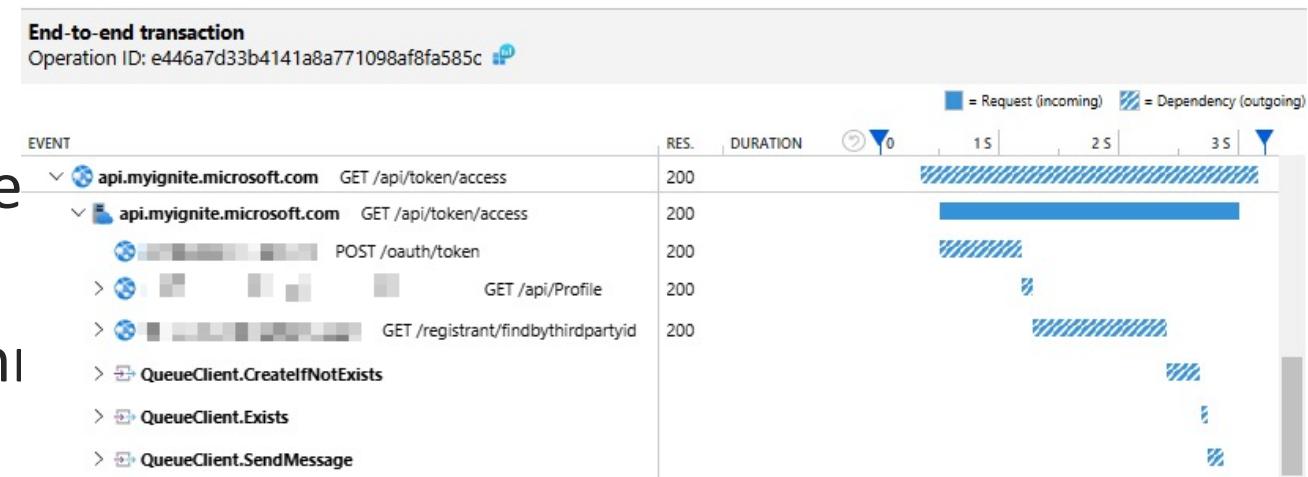


MONITORING



Capture telemetry

- Add Application Insights
 - It only takes a few clicks
- Review exception logging
 - Log errors from the browser too!
- Separate telemetry by environment
- Take care not to log PII
- Watch out for throttling
 - Adjust default config

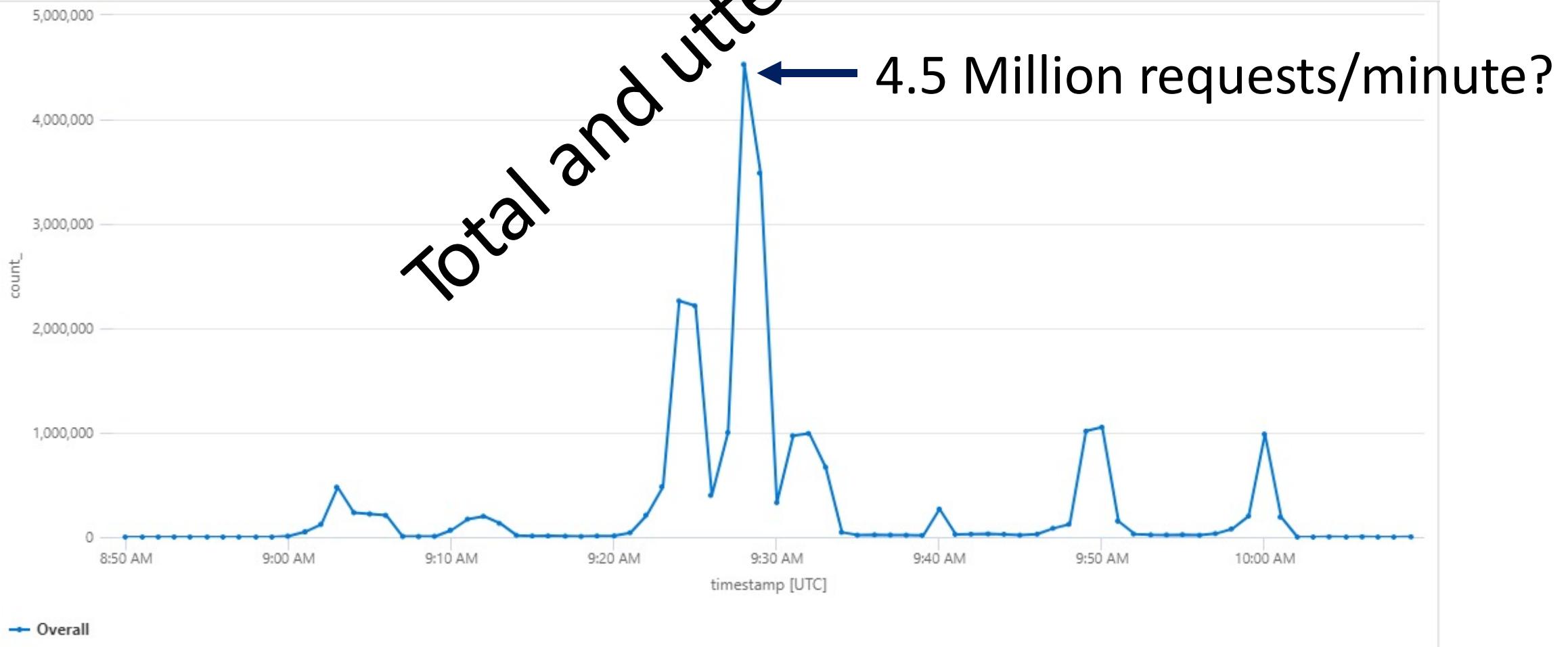


```
4 requests
5     // additional filters can be applied here
6     | where timestamp > start and timestamp < end
7 | summarize count_=sum(itemCount) by bin(timestamp, 1m)
8 | extend request='Overall'
9 // render result in a chart
10 | render timechart
```

Results **Chart** | ⏱ Display time (UTC+00:00) ▾

Completed

⌚ 00:01.5 📄 80 records





Ӣ ѿЖ҃҃čА̄ м̄т м̄тА̄г̄ ѿЖ̄ӣ ј Ж̄уТ̄ǣ, А̄ ј А̄: ѿЖ҃҃čА̄ ǣ Г̄гÓК

Ӣ , Ј̄ ј Ј̄уТ̄г̄ т̄ м̄Т̄АЛ̄ " " őJ̄ ј Ј̄уТ̄ӣ Т̄г̄J̄Ḡ ј őП̄А Ӣ ј А̄ ј Ј̄уТ̄г̄æÓК̄

g̊eJ̊ōč̊ Ä s̊t̊Ä W̊i Äg̊æ č̊č̊ ōj̊i j̊ J̊uT̊i T̊g̊J̊G̊i j̊ g̊i ÄoÄt̊ Äj̊ Cæ̊ J̊ř̊ j̊ J̊uT̊g̊óK̊

ÅÅṁęÄi ÄT̄ ūgōÅńLj i m̄ ęgj ūt̄ J̄B̄Ac m̄g Äj j J̄T̄Ḡgæ

И „, Ј; ўТѓЈГ ёЈӦčӐ т

i ūT̄ḡJ̄Ḡ, j J̄uT̄aB̄ḡ, eōj i ĀoĀt̄ Āj̄, C̄T̄J̄T̄Ōā ĀoĀt̄ Āj̄, C̄ō ūl Āgḡȳ ; t̄, J̄T̄: eJ̄ōč̄Ā K̄

Ј; ѕТѓЈГ ёЈőčӐ аёѓӐ ѕ"ј ЈИӐП т' ѕЈТԌе

¶ Pi ÄöÄł Äj, Ci j Äł gøoÄ TłAi üTçgømVWç

čí

Î· ŸTÀ ÏG e ÎÕČĂ ře e ÎÕČÁÍK

ĀĀm ččm̄i ī Ā m̄Ā iū Āqm̄ ym̄ ī Ām̄ ȳt̄i j̄t̄ā æ



Monitor performance

- Review live system under load
- Identify typical usage patterns
- Identify areas of concern
- Build dashboards to see key metrics at a glance
- Build alerts to escalate when failures occur
- Make a habit of reviewing telemetry data





Improve performance

- Review load test and live site metrics
- Perform load testing
- Run multiple load patterns
 - Ramp up, constant, soak, spike...
- Identify hotspots and big payoff improvements
- Examine profiler information
- Make a change, record, and repeat
 - Change code or platform





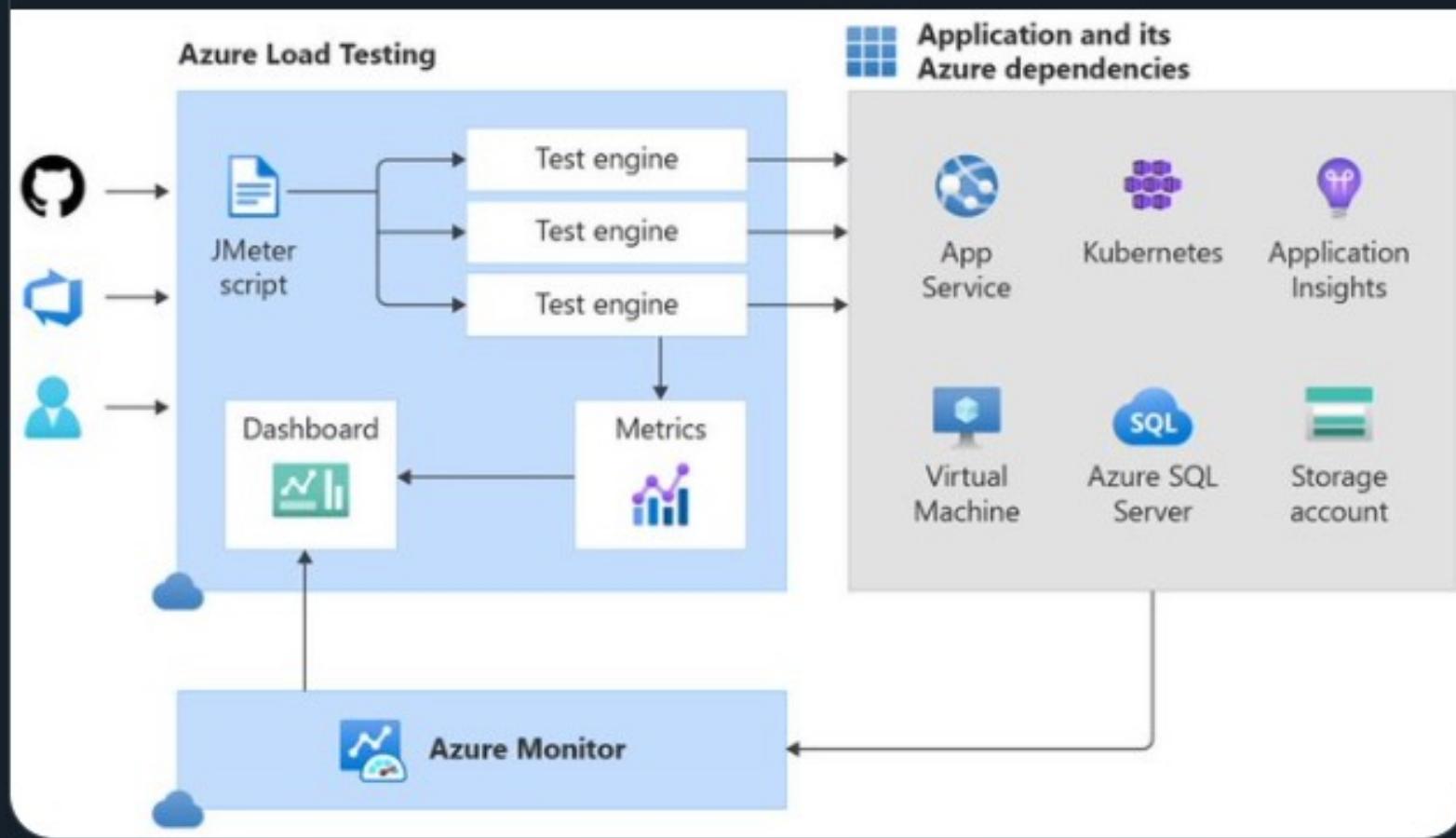
Gavin Barron 💉💉💉 @gavinbarron · 13h

Well damn, guess it's going to be a lat night for me #CloudSummit

...

📢 S:\cott Hoag @ciphertxt · 13h

Introducing Azure Load Testing: Optimize app performance at scale
dlvr.it/SDV5Zx





AZURE LOAD TESTING TOOLS



ARCHITECTING FOR SCALE

Overview of Digital Venue Architecture – Pre-COVID



Registration + User Profiles



Registration



User Profiles
REST API



SQL

Digital Venue



Website



REST API



Indexers



Cosmos
DB



Azure
Search

Sessions + Schedules



Session Manager
REST API



SQL

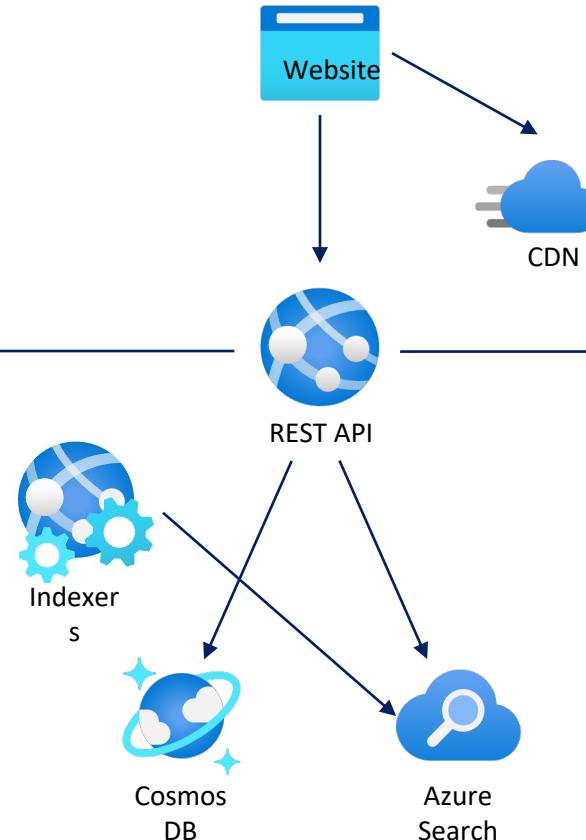
Overview of Digital Venue Architecture – Add CDN



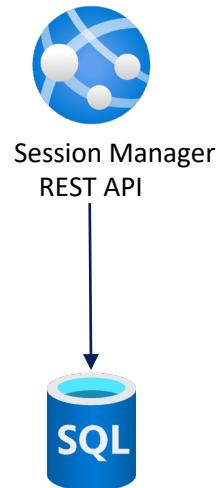
Registration + User Profiles



Digital Venue



Sessions + Schedules





Sign in process - March 2020

▼ api.myignite.microsoft.com	GET /api/token/access	200		
api.myignite.microsoft.com	GET /api/token/access	200		
▼ api.myignite.microsoft.com	GET /api/token/access	200		
> [REDACTED]	GET /api/Profile	200		
> [REDACTED]	GET /registrant/findbythirdpartyid	200		
> [REDACTED]	PUT /api/Profile	200		

Overview of Digital Venue Architecture – Early-COVID



Registration + User Profiles



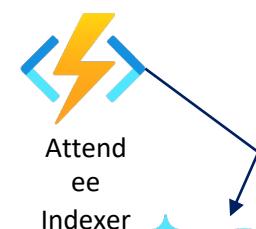
User Profiles
REST API



Digital Venue



REST API



Attendee
Indexer



Cosmos
DB



Session
Indexer



Azure
Search

Sessions + Schedules



Session Manager
REST API





Sign in process - October 2020

▼ api.myignite.microsoft.com	GET /api/token/access	200		
api.myignite.microsoft.com	GET /api/token/access	200		
▼ api.myignite.microsoft.com	GET /api/token/access	200		
> [REDACTED]	GET /api/Profile	200		
> [REDACTED]	GET /registrant/findbythirdpartyid	200		
> [REDACTED]	PUT /api/Profile	200		

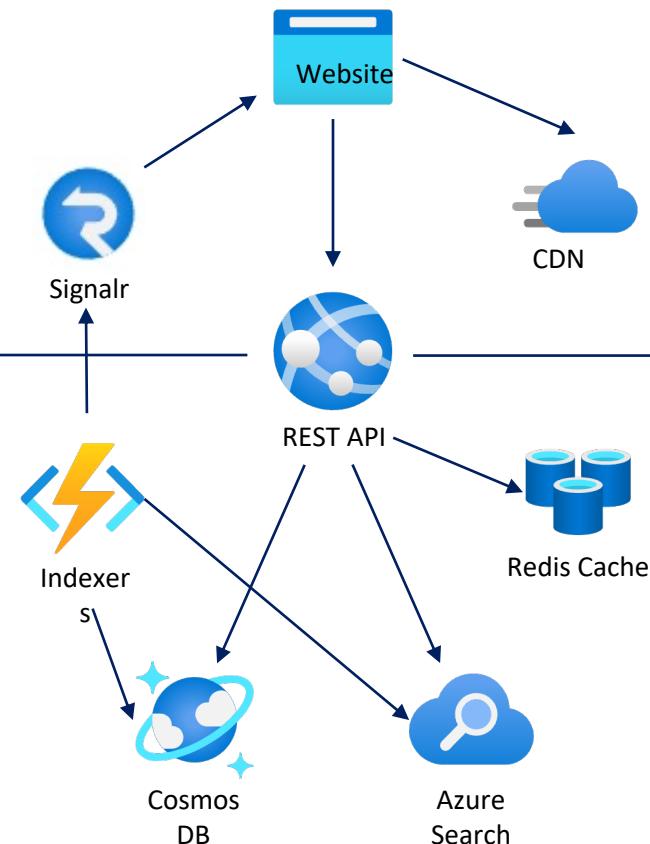


Overview of Digital Venue Architecture

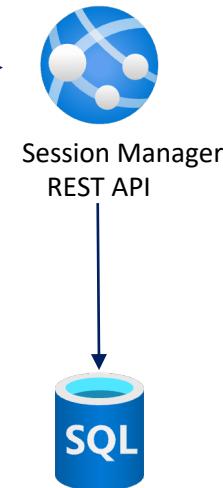
Registration + User Profiles



Digital Venue

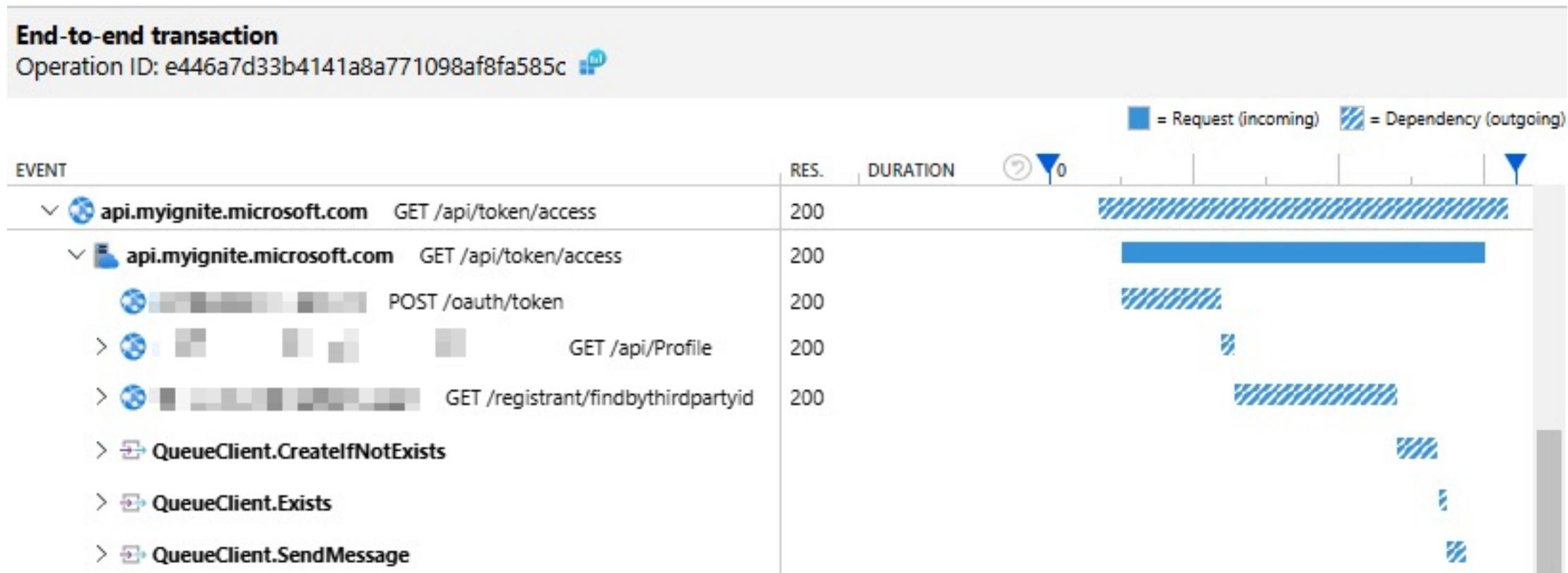


Sessions + Schedules



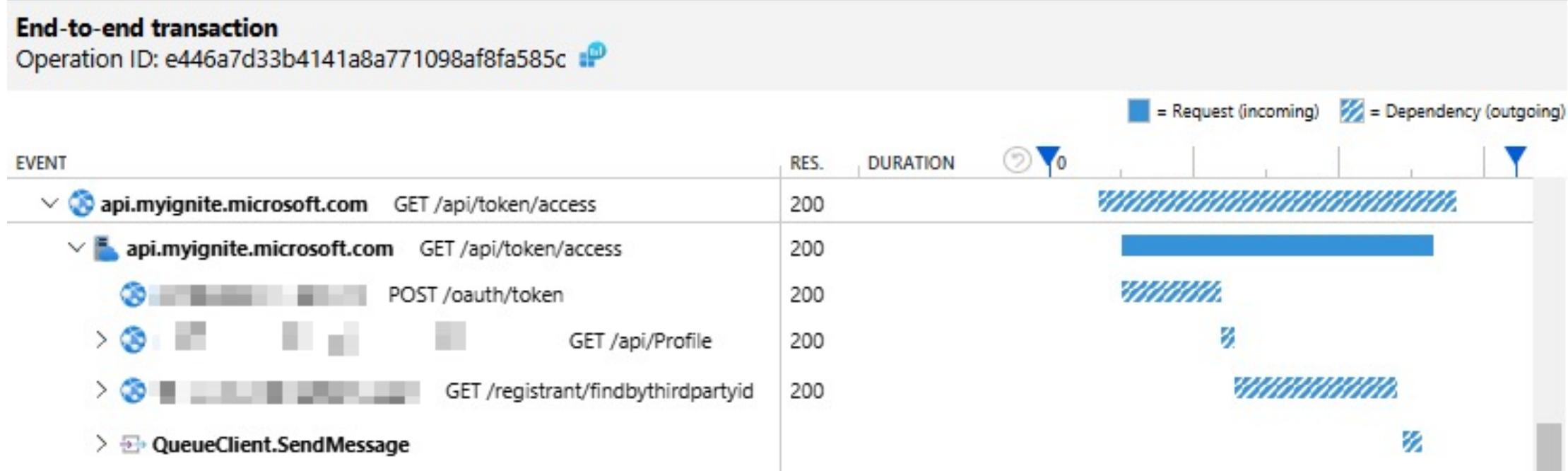


Sign in process - November 2021





Sign in process - Now





WRAP UP



Lessons learned

- As you grow in scale you will find things that stop working as well
- Your architecture may need to evolve as your system grows in scale
- Incremental improvement adds up over time
- Focus on high use areas and things that impact user experience
- Monitoring dashboards should have direct measures of user experience
- Have brown-out plans
- Ensure lines of communication are open with all stakeholders



Tools and Resources

- App Service Diagnostics: <https://docs.microsoft.com/en-us/azure/app-service/overview-diagnostics>
- JMeter: <http://jmeter.apache.org/>
- Azure Load Testing: <https://azure.microsoft.com/en-us/blog/introducing-azure-load-testing-optimize-app-performance-at-scale>
- Example JMeter scripts:
<https://github.com/ShadowPic/PublicTestProjects/tree/master/jmeter>
- Azure Well-Architected Framework: <https://docs.microsoft.com/en-us/azure/architecture/framework/>
- Azure Cost Calculator: <https://azure.microsoft.com/en-us/pricing/calculator/>



thank you

questions?



@gavinbarron



<http://gavinb.net>

