An Architecture overview of APNIC's RDAP deployment to the cloud

Presented by George Michaelson

APNIC Registry Products Team - David Holsgrove
George Michaelson
Rafael Cintra
Tom Harrison
Zen Chai

What we did

- We took an on-premises, single-VM RDAP deployment and moved it to the cloud
  - Specifically GCP, in Kubernetes
- We constructed a ”federated” model of RDAP
  - Individual RDAP instances for each source (APNIC, NIRs)
  - An ‘rdap ingressd’ process to front-end and redirect
  - CloudFlare as an RTT optimizer to select GCP region
- Why did we take this approach and what are the issues and benefits?

Problem: we have disparate data

- NIR/APNIC data sharing model is ‘loose’
  - Some whois data is shared as bulk exports of RPSL -Daily dump cycle
  - Some whois data is shared by Whois NRTM protocol -Live update
  - Some information is not currently shared
    - This is work-in-progress
    - Includes i8n tagged information in local-language
  - Some NIR operate inside APNIC processes
    - They publish in APNIC whois and RDAP directly
- How can we systematize this information management?
  - Can we ‘do better’ in RDAP?

- Problem: APNIC RDAP does not adequately reflect ’most specific’ data
  - Reports ‘held by NIR’ for significant amounts of resources

Problem: we rely on traffic to Australia

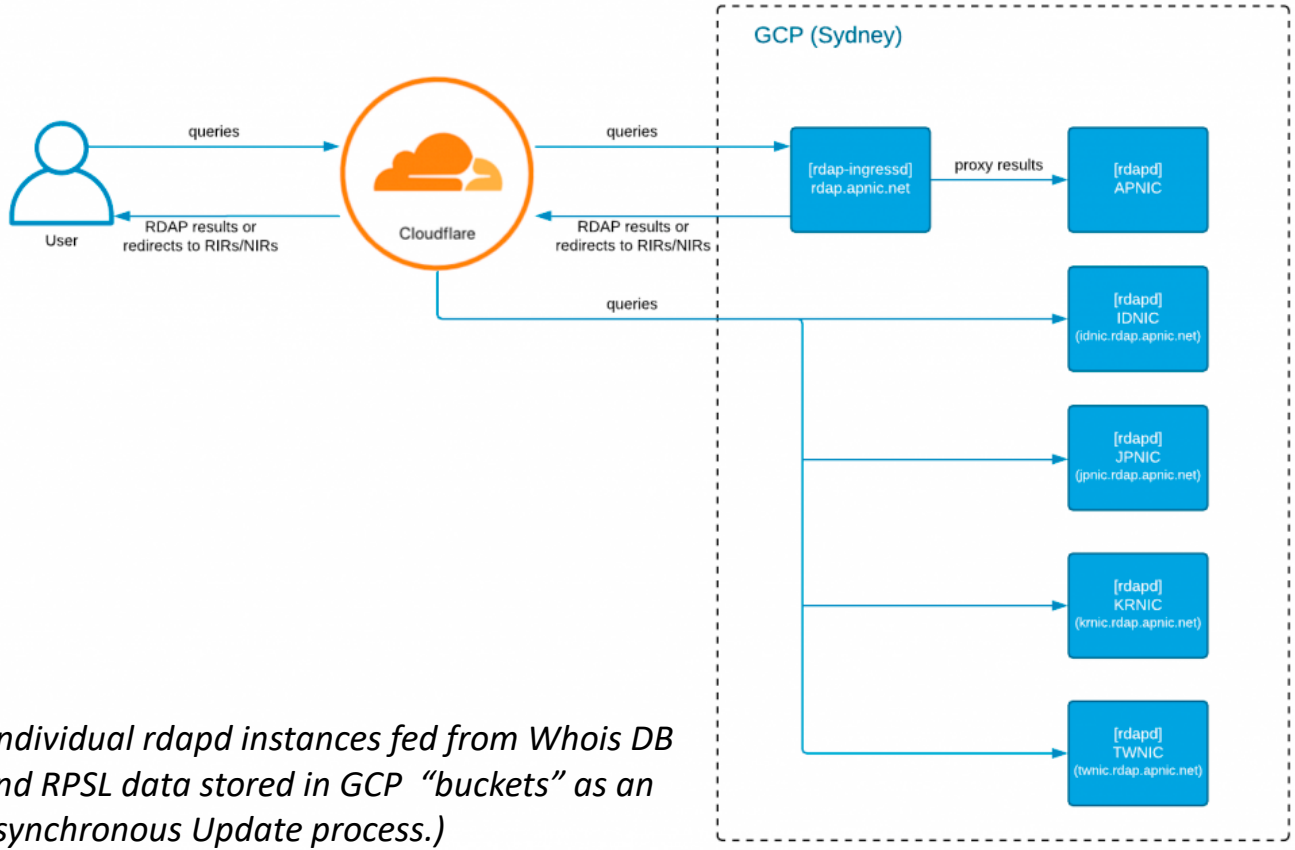
- Our previous RDAP solution operated from an on-premises VM platform
  - Initial client location survey suggested most traffic sources offshore in America and Europe at significant RTT cost
  - The right cloud deployment could significantly reduce RTT
- On-premises capital investments reaching renew time
  - Can we achieve capex/opex cost savings in cloud deployment?
    - Yes, this is just cost-shifting but for the right benefits..
- Single-Point-Failure risks
  - No redundancy against DDoS on path, or host failures
- Cloud solutions look promising
  - Distribution models come for small extra investment in the model
  - Most cloud providers in the same 15+ locations worldwide

Federated data model

- APNIC is the Asia-Pacific Regional Internet Registry (RIR)
  - 7 National Internet Registries (NIR) operate inside our model
- Loose technical coordination:
  - Not all NIR ready to deploy RDAP
  - Some NIR host & operate directly in APNIC for some services
  - Some NIR have significant on-prem investments including ccTLD function and will have in-house RDAP
- Proposed solutions for RDAP for all APNIC need to respect NIR data management differences

Solution: federated server model in cloud

- Run a front-end which can direct query to most specific data source
- Uplift Whois data in disparate methods, present single RDAP information model across all data
  - But identify specific NIR RDAP server ‘owning’ the data
- Deploy front-end traffic direction to leverage multiple-point cloud deployment



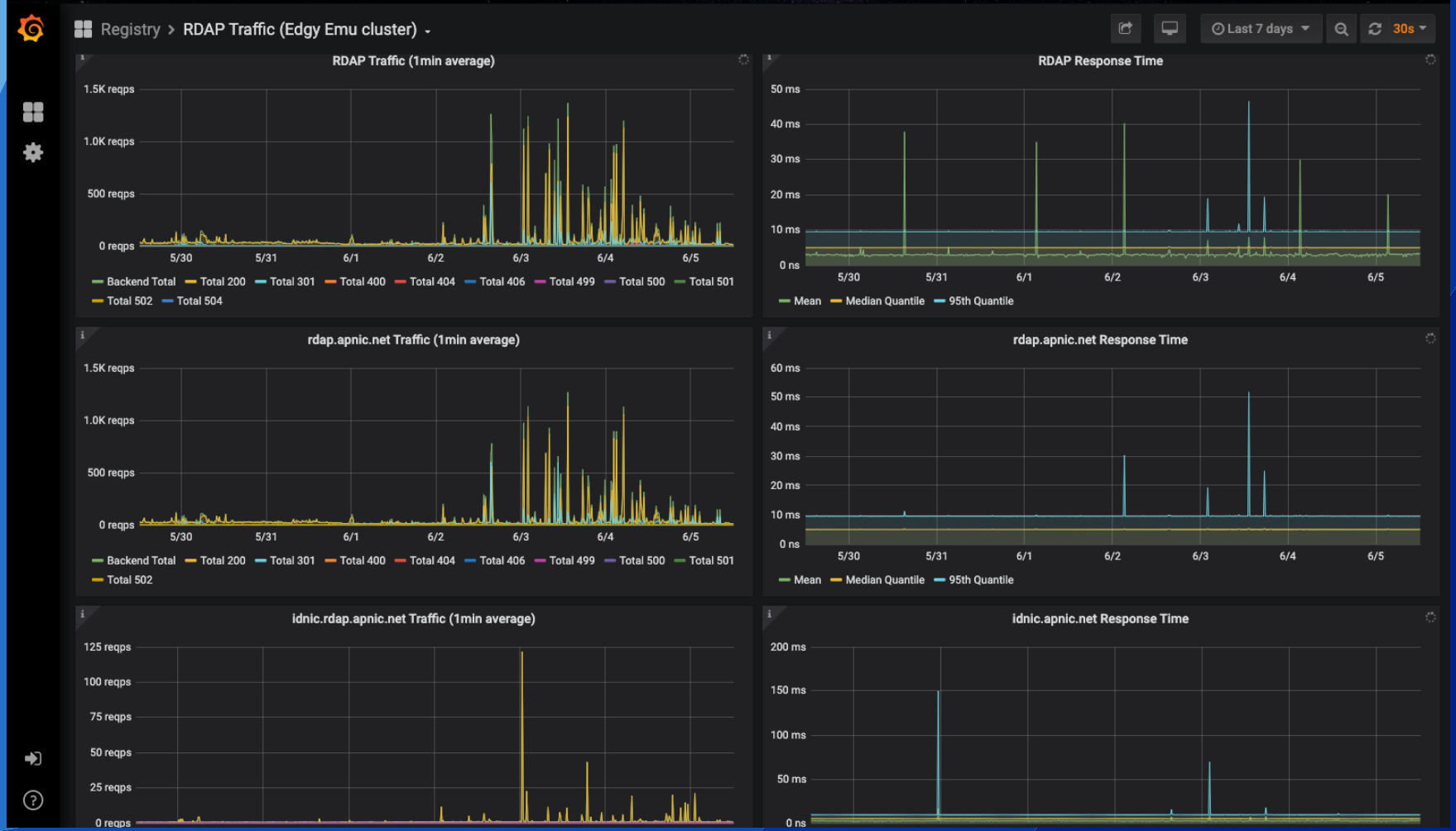
(Individual rdapd instances fed from Whois DB and RPSL data stored in GCP "buckets" as an asynchronous Update process.)

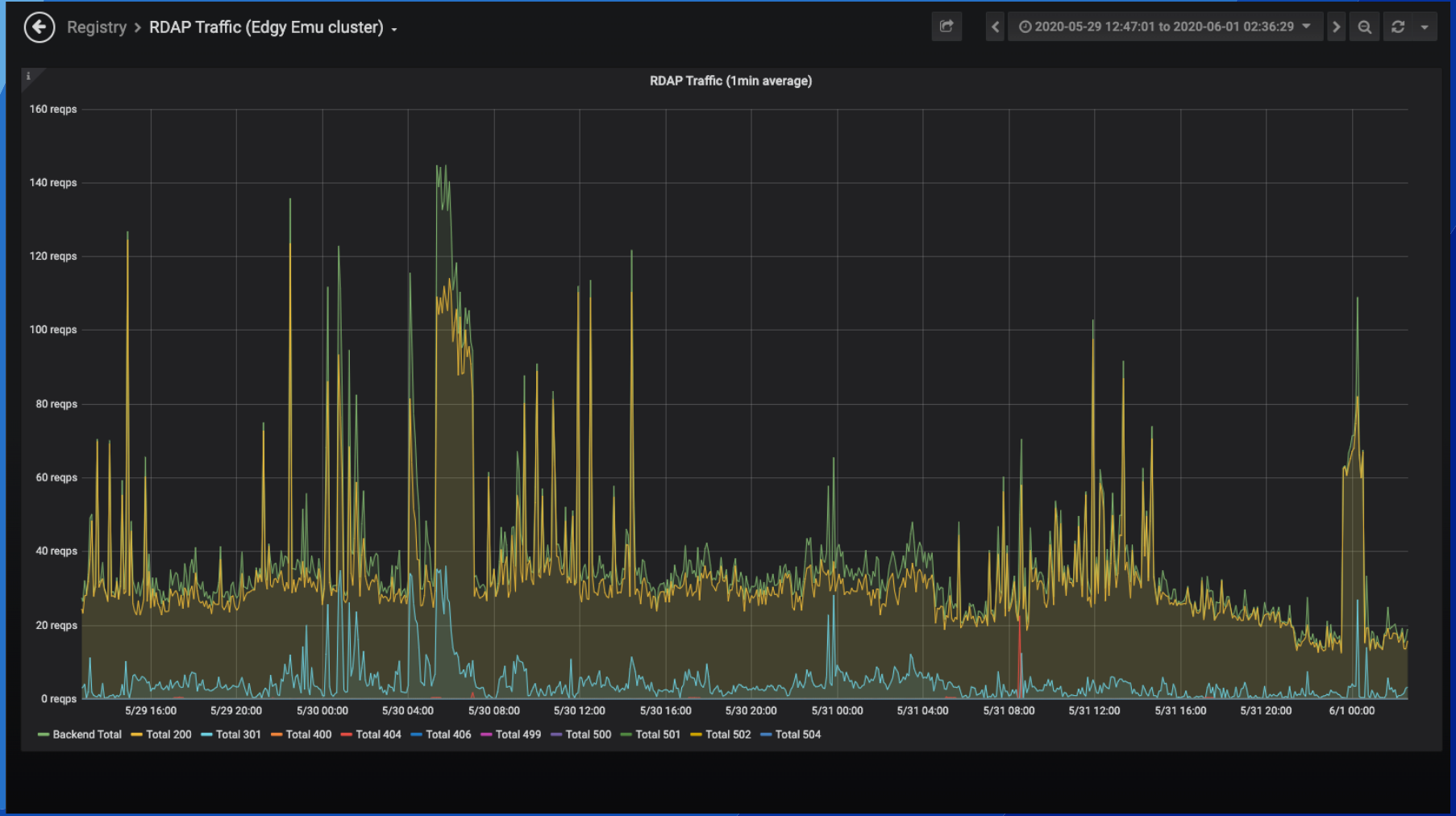
The moving parts

- rdap-ingressd acts as front-end and receives all otherwise unqualified queries
  - rdap.apnic.net -> cloudflare -> rdap-ingressd
    - rdap.<nir>.apnic.net for each NIR with individual data
- individual RDAP daemons serve data from:
  - Whois DB (RPSL converted to RDAP)
  - Google ‘buckets’ (feeds pre-converted where need be to RPSL)
    - De-duplication, source specific conversion is done during bucket insertion
- Flow control/redirection model fed from bucket
  - ‘delegated’ data to identify RIR
  - ‘delegated for nir’ data to identify which NIR has more-specific data
  - AVL held in-memory for fast lookup to most specific source for query
  - Update to RDAP service now fast for live-update sources, daily sync for others

Systems monitoring

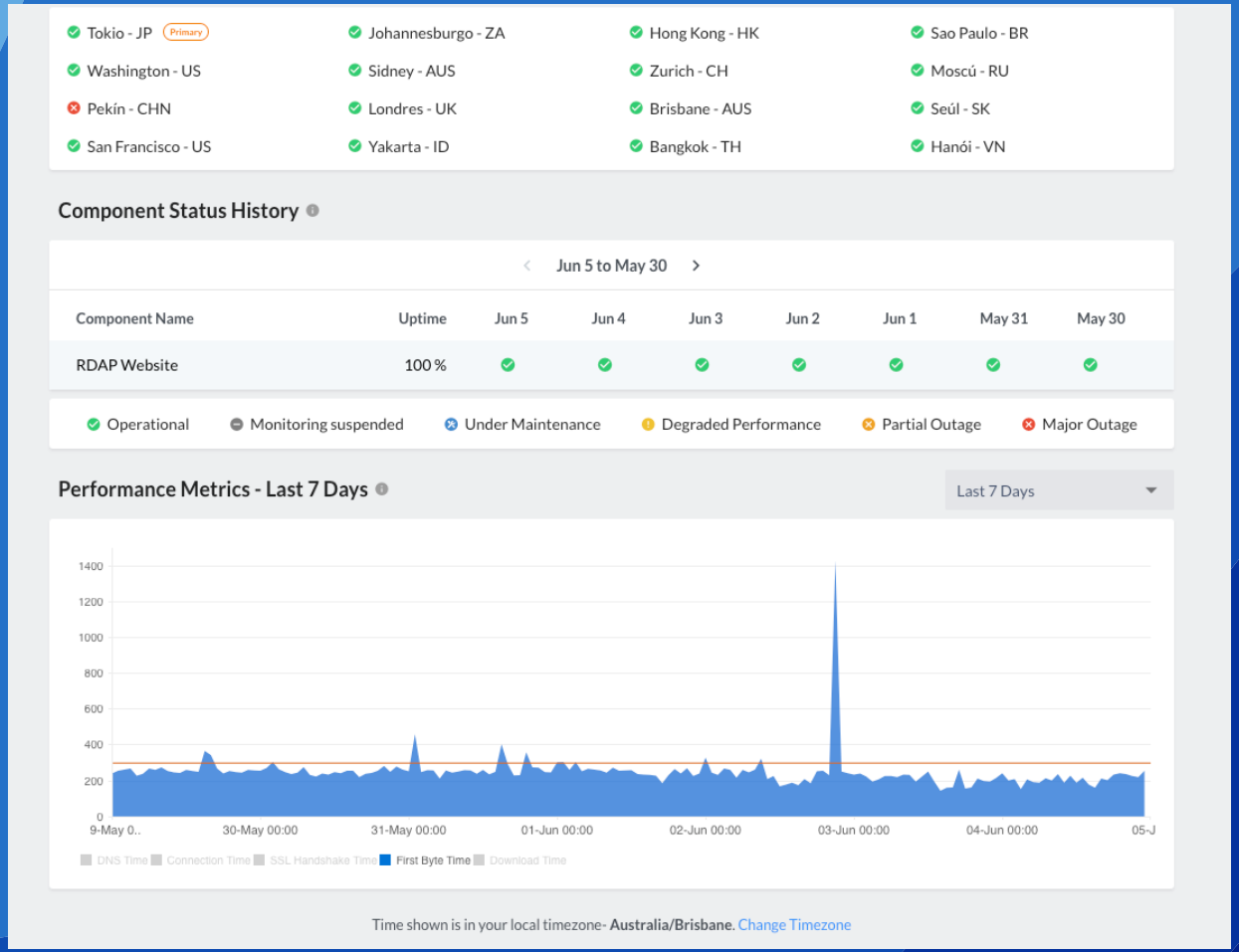
- Prometheus data feed to Grafana
  - Standard operations tools to monitor service
- External measure by site24x7 services
  - Seven points of connect worldwide including some long-delay paths
  - Sensu alerting on service outages


Registry > RDAP Traffic (Edgy Emu cluster)
Last 7 days
30s
RDAP Traffic (1min average)
1.5K reqps
1.0K reqps
500 reqps
0 reqps
5/30
5/31
6/1
6/2
6/3
6/4
6/5
Backend Total
Total 200
Total 301
Total 400
Total 404
Total 406
Total 499
Total 500
Total 501
Total 502
Total 504
RDAP Response Time
50 ms
40 ms
30 ms
20 ms
10 ms
0 ns
Mean
Median Quantile
95th Quantile
rdap.apnic.net Traffic (1min average)
rdap.apnic.net Response Time
60 ms
idnic.rdap.apnic.net Traffic (1min average)
125 reqps
100 reqps
75 reqps
50 reqps
25 reqps
idnic.apnic.net Response Time
200 ms
150 ms
100 ms


Registry > RDAP Traffic (Edgy Emu cluster)
2020-05-29 12:47:01 to 2020-06-01 02:36:29
RDAP Traffic (1min average)
160 reqps
140 reqps
120 reqps
100 reqps
80 reqps
60 reqps
40 reqps
20 reqps
0 reqps
5/29 16:00
5/29 20:00
5/30 00:00
5/30 04:00
5/30 08:00
5/30 12:00
5/30 16:00
5/30 20:00
5/31 00:00
5/31 04:00
5/31 08:00
5/31 12:00
5/31 16:00
5/31 20:00
6/1 00:00
Backend Total
Total 200
Total 301
Total 400
Total 404
Total 406
Total 499
Total 500
Total 501
Total 502
Total 504

Did we over-cook things?

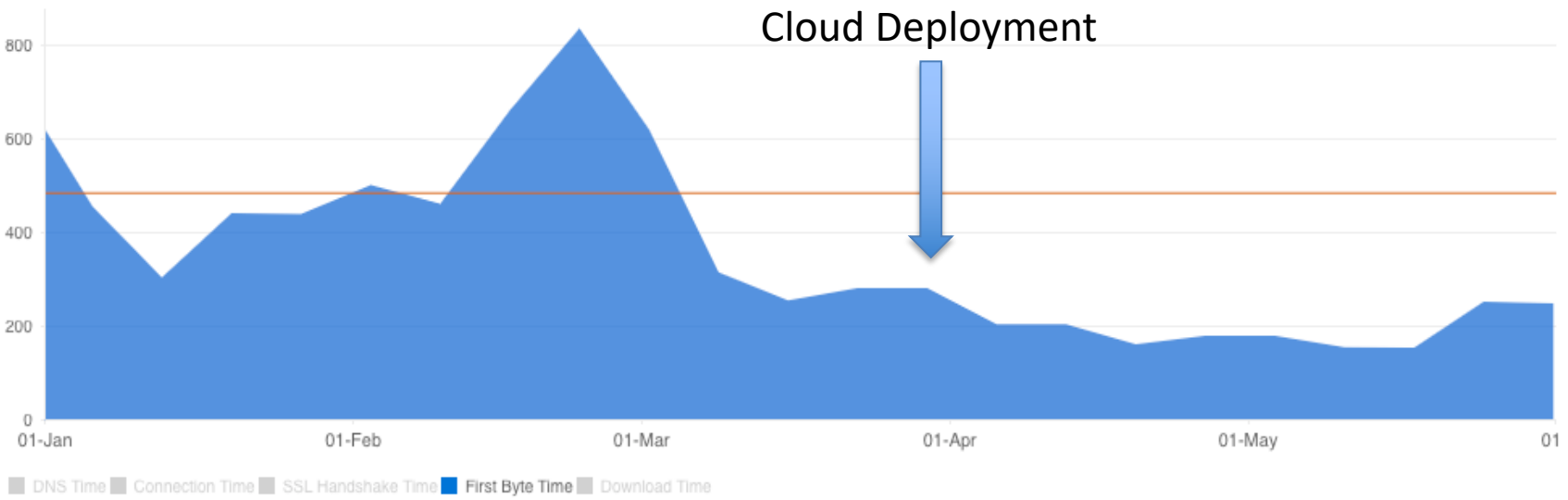
- Kubernetes added complexity
  - Its flexible, but it also has many moving parts.
- Strictly speaking we can handle load from a single site
  - The real benefit to us in cloud, is achieving lower latency
  - There are benefits in pod liveness/horizontal scaling under load
- What we gained from this deployement is flexibility
  - We can handle future changes including redirection to the NIR
  - Our Kubernetes cloud investment is going to host other services



- 16 points of check worldwide
- (China higher delay/availability)
- Stripchart shows time to first data byte
  - 200ms normal

Performance Metrics - This Year

This Year



Time shown is in your local timezone- **Australia/Brisbane**. Change Timezone

Outcomes: lower RTT and higher availability

- Pre-deployment RTT variant 200ms-500ms
- Post-deployment consistent 200ms RTT
  - (with thin spikes in site24x7)

- Pre-deployment APNIC only data ‘resource in NIR’
- Post-deployment: more specific data for 4 NIR
  - IDNIC JPNIC, KRNIC and TWNIC
    - IRINN/VNNIC data maintained in APNIC
    - CNNIC data pending Whois RPSL feed

Benefits

- Clean separation of APNIC & NIR managed data
  - Easier to understand data management obligations and design for migration of service back into NIR
- Pod scaling can include more specific focus
  - Scale Japan for high JP directed query load
  - Better cost controls
- Federation can include uplift of new NIR into our cloud
  - We can offer the NIR identical CDN distribution benefits
- Rdap-ingressd can ’30x refer’ out to the NIR
  - We can support in-house NIR RDAP deployment if they prefer
  - RTT no longer bound to 2x RTT inside the POD costs

Issues

- Serving a query can incur two HTTP round-trips to select the right RDAP instance inside the pod
  - Subsequent queries can go direct where URL is passed
  - Potential mitigation: CloudFlare “worker” URL processing
- Horizontal scaling delay for large footprint processes
  - Spike duration has impact on benefit of scaling, need to oversize live instance for short-duration peaks
    - Potential mitigation: can we design lower deployment time processes?
- Depends on a Whois DB replica per-pod
  - Potential improvement: RDAP mirroring protocol

Whats next? Distribution!

- Using existing Helm/Kubernetes configuration build out second (and future) nodes
  - Potential to reduce per-node sizing for cost saving
- Second deployment targeting Tokyo
  - Good transits to Asia-Pacific (1 ocean hop in many cases)
  - Incrementally better for US, Europe
  - We expect RTT to reduce significantly for Asia, US and Europe
- Future nodes: US/Europe
  - Higher resiliency. More potential for node sizing reduction

Whats next? The potential mitigations

- CloudFlare “worker” URL processing
  - Demonstrator work-in-progress. Uses delegated map as index/trie to most specific source.
  - Removes many 2xRTT 30x re-query delays
    - Will redirect to other RIR as well as specific NIR inside APNIC pod
    - Will need to synchronise updates in Cloudflare worker with data changes inside the RDAP pod
- Can we design lower deployment time processes?
  - Less memory intensive state? Buckets? (increases RTT)
  - Has large cost saving potential
- RDAP mirroring protocol
  - Remove the in-Pod whois dependency

See
https://blog.apnic.net/2020/04/01/a-new-infrastructure-to-serve-rdap/

QUESTIONS?