

Who are we?



Pavel Nikolov

Principal Software Engineer

pavelnikolov

Matt Hope

Principal System Engineer

matthope

Backstory

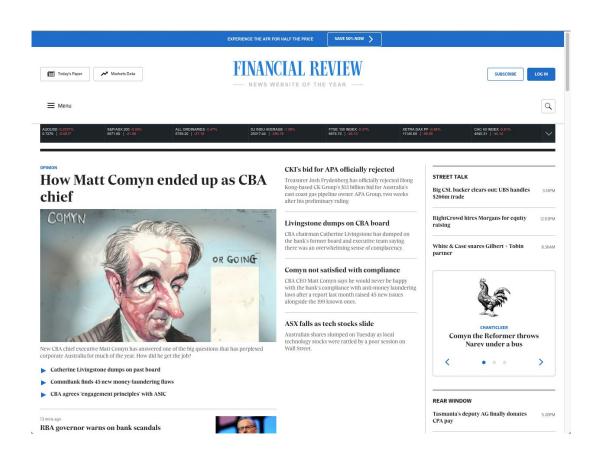


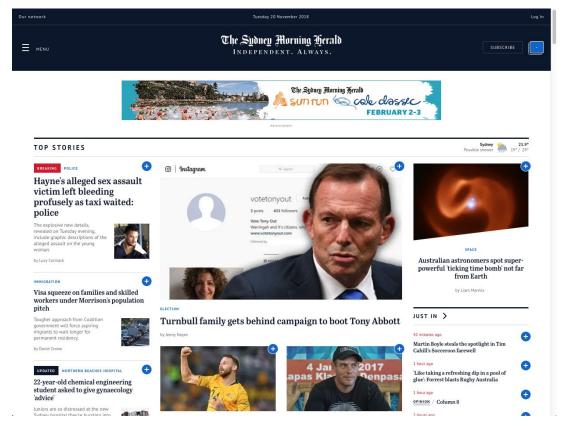
- We work in a 180-year old media publishing company
- Change from within is very difficult; on the other hand
- Media has zero barriers to entry (now there is more competition than ever)

Some of our mastheads









How is this relevant to you?



- You are in one of these situations you are:
 - considering a cloud native transformation,
 - about to begin,
 - in a transition process; or
 - already reaping the benefits of the transformation

Our Journey







Planning





North America 2018

PLANNING



































Corporate Reorganisation



Few important things happened at the same time

- Corporate reorganisation
- Rewrite of almost all our systems (websites, mobile apps and backend systems)
- Cloud native transformation

Inception







The Start of the Transformation



- We set up a Kubernetes cluster; and then we
- Started exploring and experimenting with it
 - This didn't continue for long because we had tight deadlines...
- It was high time we setup our real cluster(s)

The Unknown







Where do we start from?



- With no prior Kubernetes experience at enterprise scale; and
- No prior microservices experience in the team
- There were many (unknown) unknowns...

Preparation







Important Key Decisions



- Set up different Kubernetes clusters per environment
- Setup a namespace per application per environment
- (Almost) all our applications use the same helm chart
- Have a CI/CD and deployment process

Skeptic







"No" is temporary; "Yes" is forever



- We've been very skeptical when introducing new tech to our stack
- Resisted using Linkerd or Istio in the early days
- Same goes for serverless and other technologies
- What we learnt:
 - Be careful when introducing a new technology! And only do it if it's absolutely necessary!

Secret





North America 2018 -







Think Secrets First



Another advice from us:

- Start with your secrets solution from day one! We can't stress this enough!
- Before you deploy any code to prod try to:
 - Use, rotate and revoke secrets

Forbidden







Don't leave access control for later



- Regardless of your company size or use cases, make sure you have an access control plan from the beginning
- Security is a fundamental step during the design phase of every system
- Security and access control is hard to retrofit

North Observability











Setup Observability Infrastructure



- Your immediate next step is to ensure good observability infrastructure
- What we mean by observability:
 - Centralised logging (preferably with asynchronous processing)
 - Metrics (Prometheus)
 - Tracing (OpenTracing / Jaeger)
 - Alerting (Prometheus + 3rd party services)

South Observability





North America 2018

PLANNING











Observability is key!



Once the infrastructure is in place you have to ensure that:

- All your applications use a levelled and structured logger
- Send tracing data
- Export application metrics
- Export system metrics
 - Link metrics to k8s resource/limits/HPA

Micro







Do you speak microservices?



- Does your team have any prior experience with microservices?
 - Ours did not...
 - We had to learn a lot on the fly
- It might not be immediately obvious but microservices are related to multiple CNCF projects and you better be prepared

Serialisation







Don't just use JSON



- Early on we decided not to use JSON for message serialisation for service-to-service communication
- Instead we use gRPC in (almost) all of our services

The Skeleton







The (app) Skeletons



When you build a new system of microservices from scratch you generally have two options:

- Use a microservice library; or
- Create an application skeleton (boilerplate)
 - This ensured that all our apps use metrics, tracing, proper logging, same helm chart, same CI configuration, health/readiness probes, gRPC, useful middlewares, infrastructure code and more

Knowledge







Knowledge Sharing



In order to avoid silos and ensure that everyone on the team has enough knowledge about everything we organize:

- Regular "brown bag" sessions
- "Level up" sessions
- Trainings
- Time for experiments

We also try to get involved with open-source as much as possible

DevOps







DevOps culture



For all apps we try to apply the following rules:

- You build it, you own it
- Developers from each team get paged if their apps are not behaving well
- Platform engineering team gets paged only for system-wide issues
- ChatOps
- GitOps
- Strict pre-deployment checks

Business







Benefits for the business



- Our culture changed significantly
- Lead time to feature delivery decreased from months to days
- More frequent production releases
 - before once every 3 weeks
 - after a dozen times per day
- The teams are much more motivated now

There are more islands to visit



It's a journey, not a destination!





North America 2018 ———

Questions?

