微服務化的 K8S 修鍊之路

91APP技術總監 / 劉峰全 Timothy Liu

大綱

- 91APP 簡介
- 回顧 -- 仍在進行式
- 推行 DevOps & SRE
- 微服務化遇到的問題和實作
- 其它 (應該沒時間...)

Yes, 我願意分享這個簡報

91APP

台灣最大&成長最快新零售服務商

- 2013年成立
- 超過20年零售及電商豐富經驗
- 前Yahoo!、興奇科技經營團隊

最佳商業模式
品牌新零售解決方案

- 快速建構eCommerce、O2O、Omni-Channel CRM、零售 AI 相關應用及整合數位媒體行銷等
- 連續三年榮獲「創新商務獎/最佳商業模式」
- 獲選「勤業眾信亞太區高科技高成長前500強」
  (Ranked 152th,Deloitte Technology Fast 500 Asia Pacific)

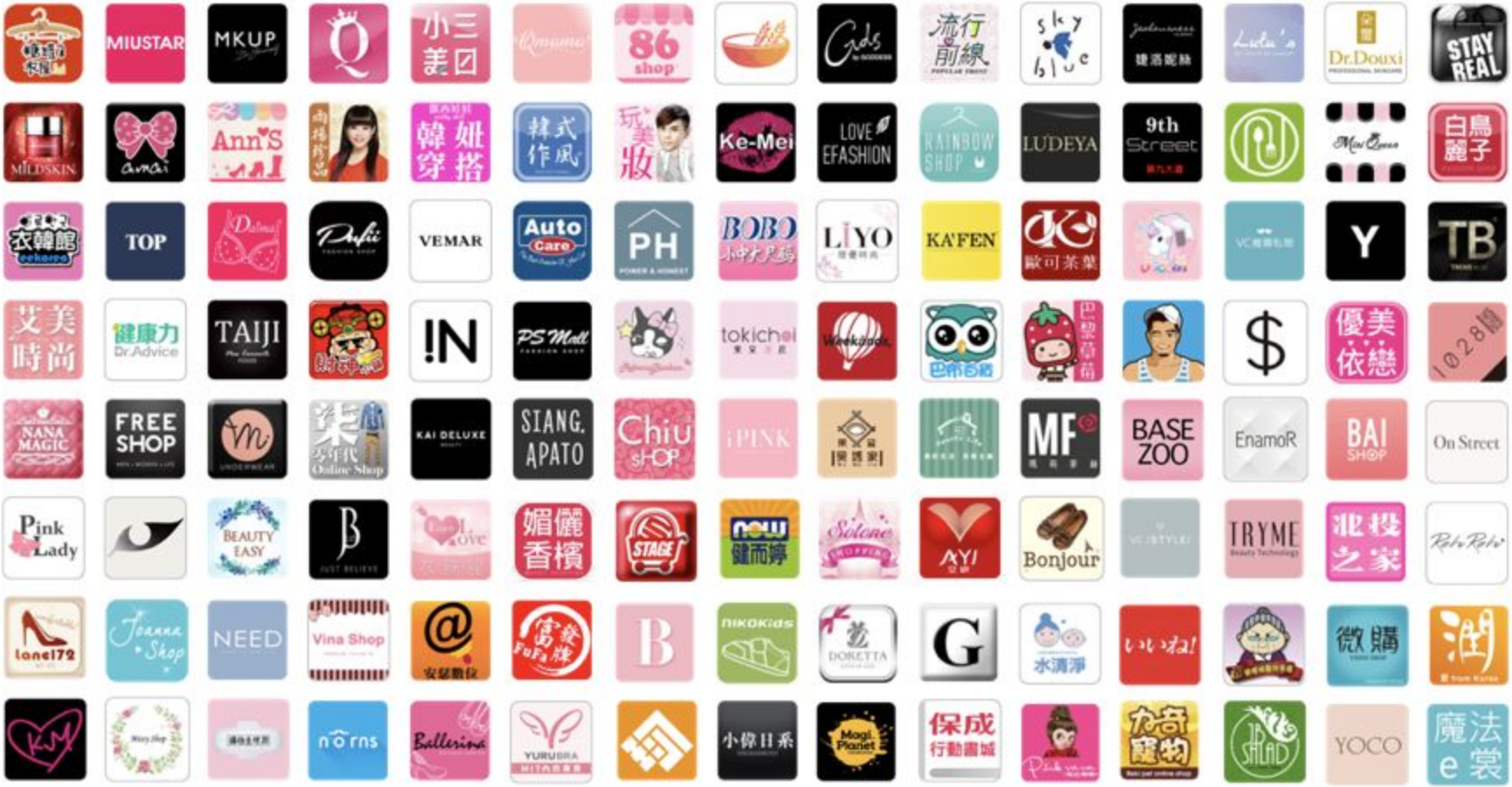
總部台北 / 320人

國內外知名實體零售品牌青睞

91APP

獲得超過10,000家品牌客戶肯定

91APP



個人簡歷

- 現任職 91APP，負責雲端系統架構與微服務規劃之 PoC 與導入
- 開發或輔導上線到 K8S 的微服務已有 6 個 (2017~)
- 1st 夢想 - 寫出能幫忙寫程式的程式
- timothyliu@91app.com

您還沒有逛過任何商品!

快來逛
人氣商品推薦

全站商品
人氣商品推薦
★超值魅力單品★
★最新單品★

加入會員送100元線上購物金

SO NICE

商品分類 最新商品 人氣熱銷 門市資訊

APP下載 | 我的帳戶 | 購物車 0

搜尋

6/12-6/15 官網限定

PRETTY SUMMER GIFT

滿 $3280

贈 質感優雅吊飾

隨機出貨・恕不挑款・送完為止

輪播設定

輪播廣告　設定數量：3

Banner

依裝置上圖

上傳手機圖素

圖片尺寸：1920寬x不限高
圖片大小：需小於450K
圖片支援：jpg / png / gif

選擇圖片

上傳桌機圖素

圖片尺寸：1920寬x不限高
圖片大小：需小於450K
圖片支援：jpg / png / gif

選擇圖片

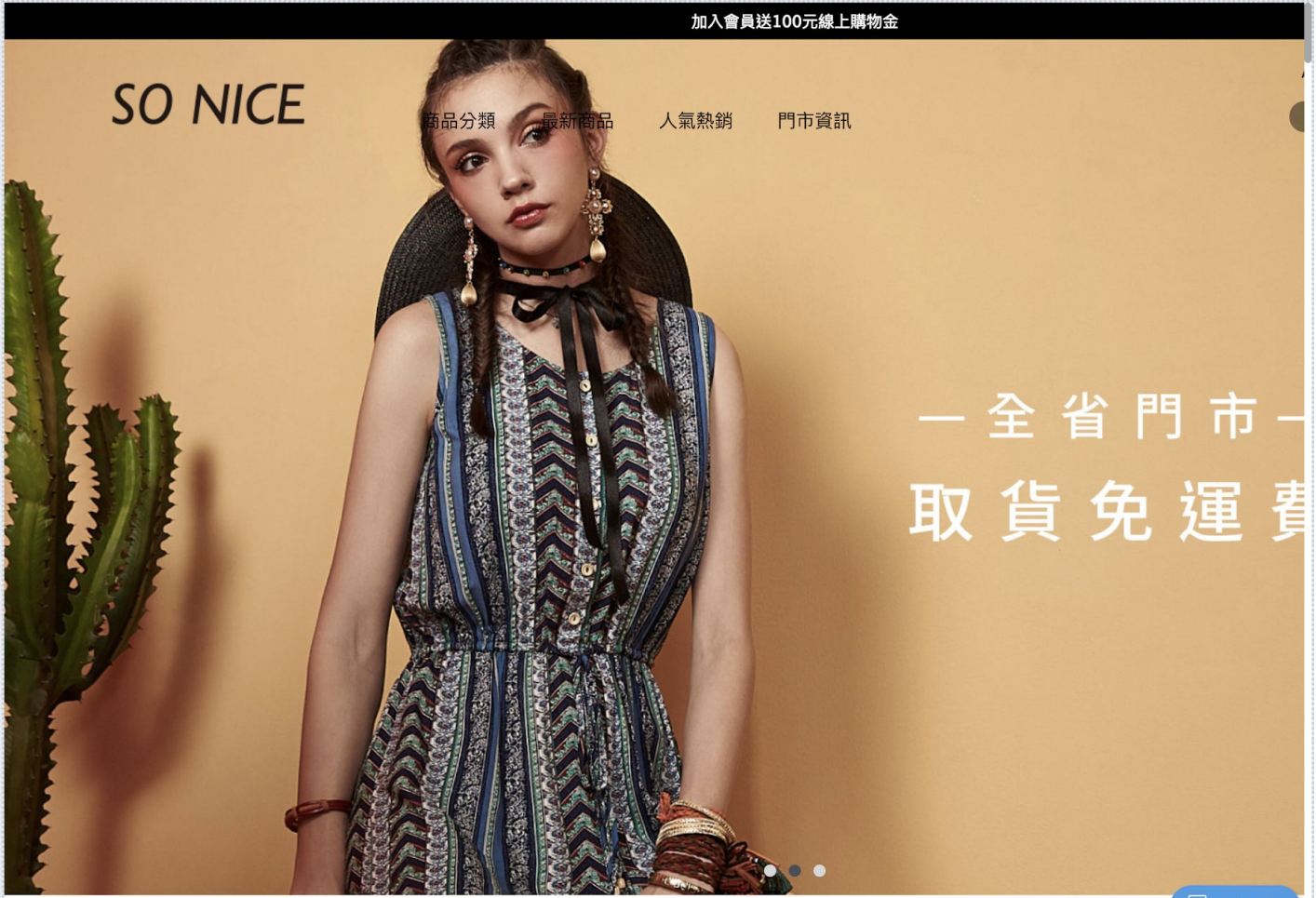
連結　https://www.so-nice.com.tw/v2/

時間排程

2018/06/11 23:45 ~ 2018/06/15 23:45

Banner

Banner



意見回饋

Header設定

Logo

SO NICE

圖片尺寸：300寬x88高
圖片大小：需小於20K
圖片支援：jpg / png / gif

選擇圖片

置頂背景

背景顏色 #FFFFFF

選單文字

文字顏色 #000000

Badge

背景顏色 #FF0000

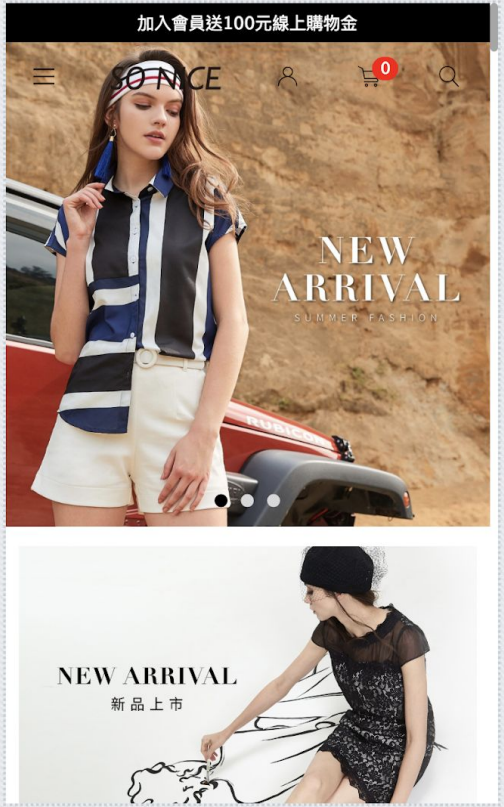
文字顏色 #FFFFFF

置頂訊息

訊息 開

訊息文字 加入會員送100元線上購物金

背景顏色 #000000



意見回饋

回顧

仍在進行式

亮光 2016 → 出發 2017

- 緣由
  - 危機→需求→挑戰→轉機→任務→行動
- 單體 vs. 微服務
  - 微服務的挑戰、Agile 組織/流程
- 選擇 Kubernetes、導入歷程
- 系統架構藍圖

Architecture Roadmap

91APP

技術 vs. 團隊

- 微服務架構
- Reverse Proxy / API gateway
- 容器技術
- 跨雲／跨平台容器管理
- Serverless

- 漸進調整和服務相對應的組織
- 開發流程 CI/CD 敏捷化
- 推動 DevOps, SRE
- T / π 型人才

微服務化的挑戰

91APP

hpa, vpa, cluster autoscaler, custom metric, livenessProbe

resource json/yaml, configmap

service (alb/nlb), ingress

nodeSelector/ podSelector/affinity taint/tolerations, custom schedule

livenessProbe, readnessProbe, cross zone, instance group (nodePool)

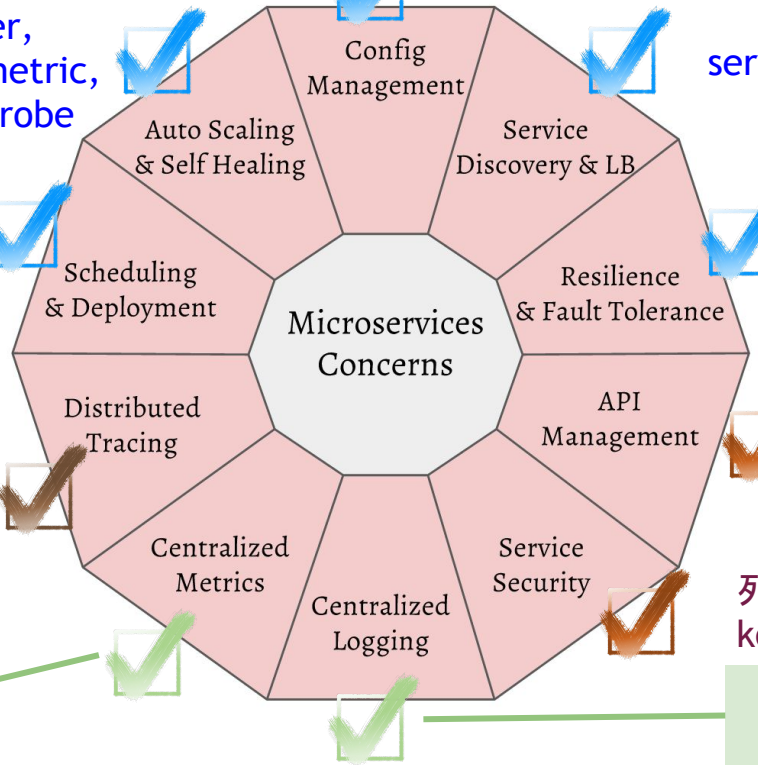
kong ingress, aws api gateway

Stackdriver Tracing / APM

fluentd + cloudwatch prometheus + grafana

列舉 port, rbac, network policy, kops, ...

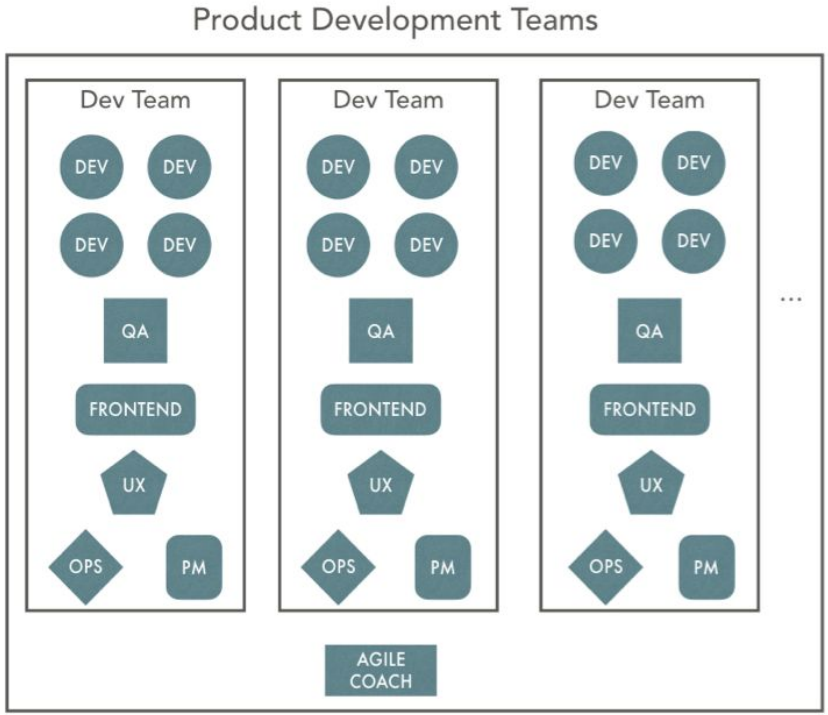
fluentd + cloudwatch / s3

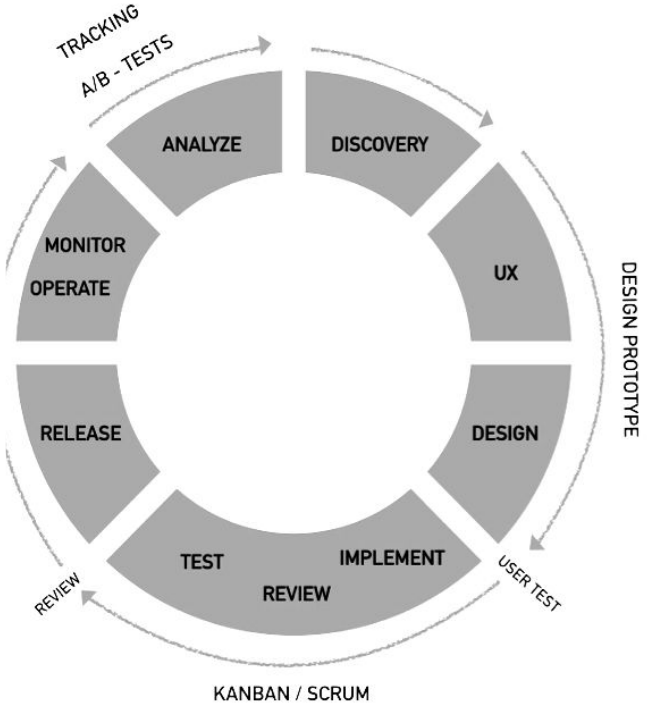


https://dzone.com/articles/deploying-microservices-spring-cloud-vs-kubernetes

Agile 組織／流程

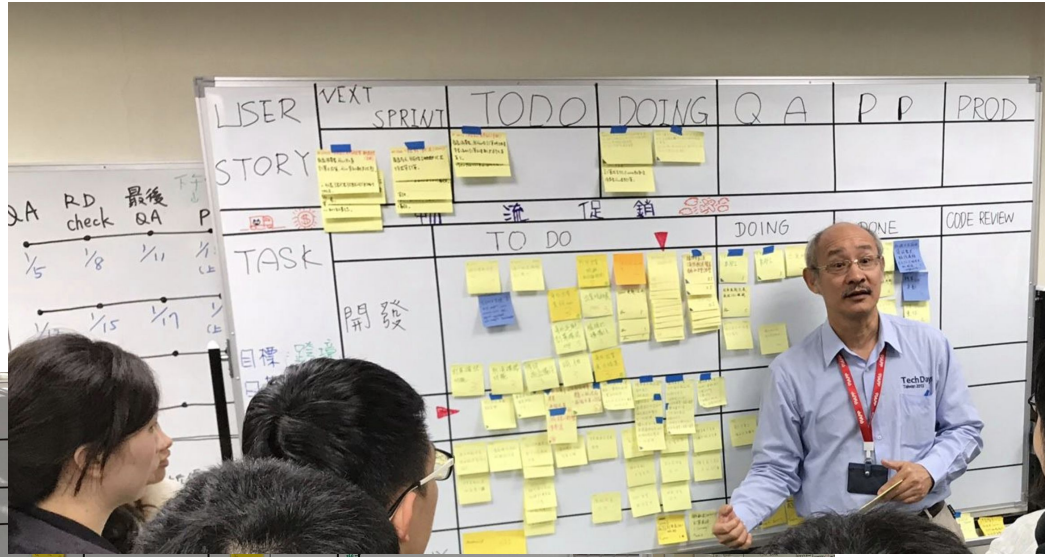
多職能團隊 + 看板 + SCRUM

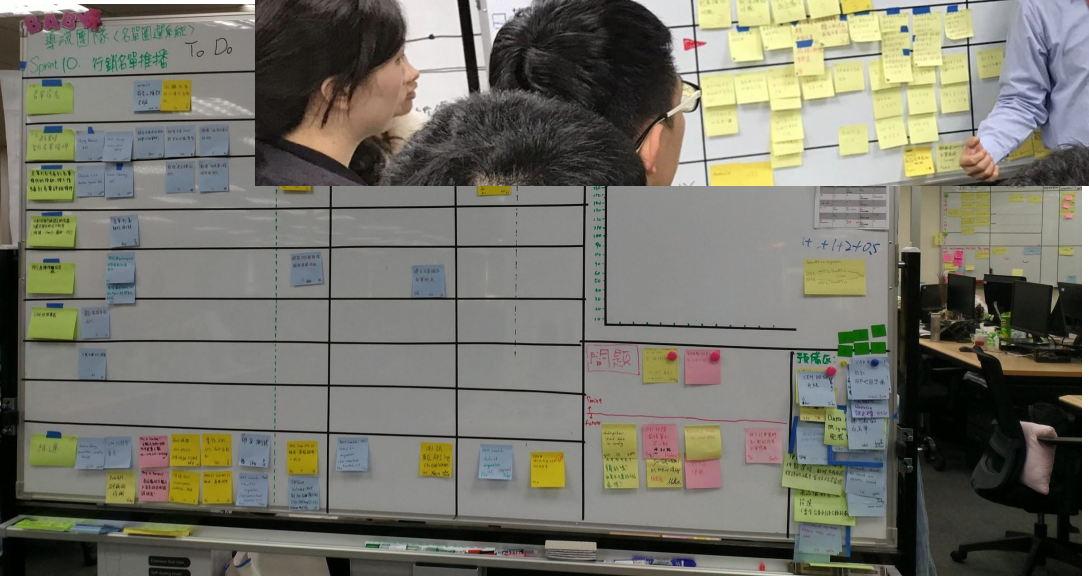




出處：http://www.infoq.com/cn/articles/agile-sailors-microservices

精實看板


USER STORY
NEXT SPRINT
TODO
DOING
Q A
P P
PROD
TASK
TO DO
DOING
DONE
CODE REVIEW
開發


To Do

學習型組織

91APP

- 2017.08 Ruddy 老師 (李智樺) 加入”總經理室敏捷教練”
  - 開始導入敏捷、持續交付和三步工作法
  - 價值流 → 看板 + VSTS → 敏捷開發
- 2017.09 全面導入 unit test，舉辦競賽，培訓專家
- 2017.10 google 協助取經 icook 愛料理 (Richard)
- 組織持續調整：市場導向團隊 + 職能導向團隊
- I 型專家 → T 型通才 → E 型 (π)
  - 2018.05 Andrew Wu 吳剛志擔任架構師 (安德魯的部落格)

DevOps

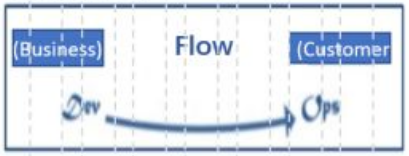
Developer + Operator = DevOps?

第一部分 DevOps 介紹

三步工作法

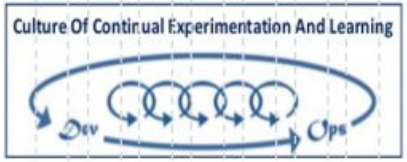
流. Flow

By: Gene Kim



回饋. Feedback

文化. Culture



技術實踐
三步工作法

透過看板驅動 DevOps 快速交付價值

https://ruddyblog.wordpress.com/2017/10/28/三步工作法思維導圖/

學習型組織

91APP

學習二週後，尚能記得的比例%

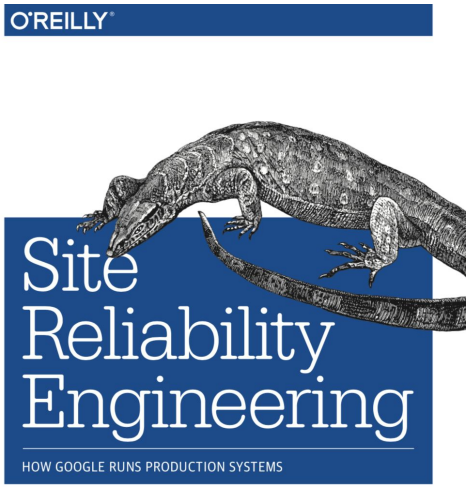
Lean Coffee

Code Review

91APP

SRE

Site Reliability Engineering
網站可靠性工程



Edited by Betsy Beyer, Chris Jones,
Jennifer Petoff & Niall Richard Murphy

When DevOps meets SRE

91APP

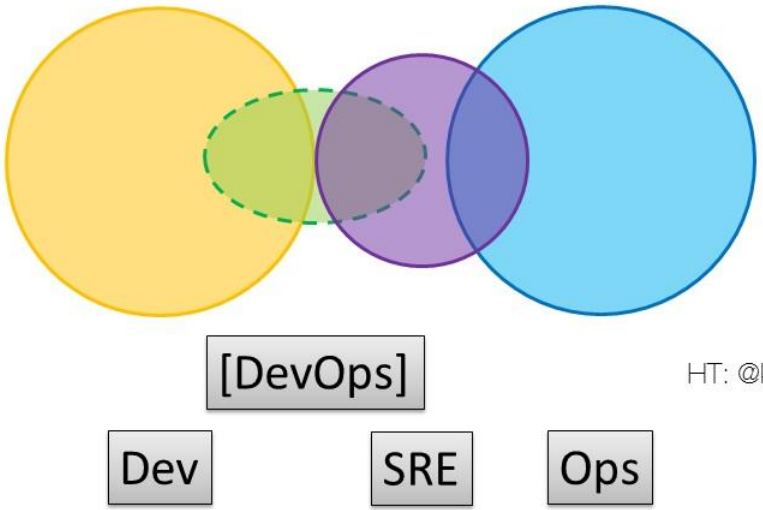
https://www.slideshare.net/sanjeev-sharma/from-apollo-13-to-google-sre

- DevOps: **“Everyone is responsible for delivery to production.”**
- SRE: **“(Everyone) is responsible for delivering Continuous Business Value”**

class SRE implements DevOps

91APP

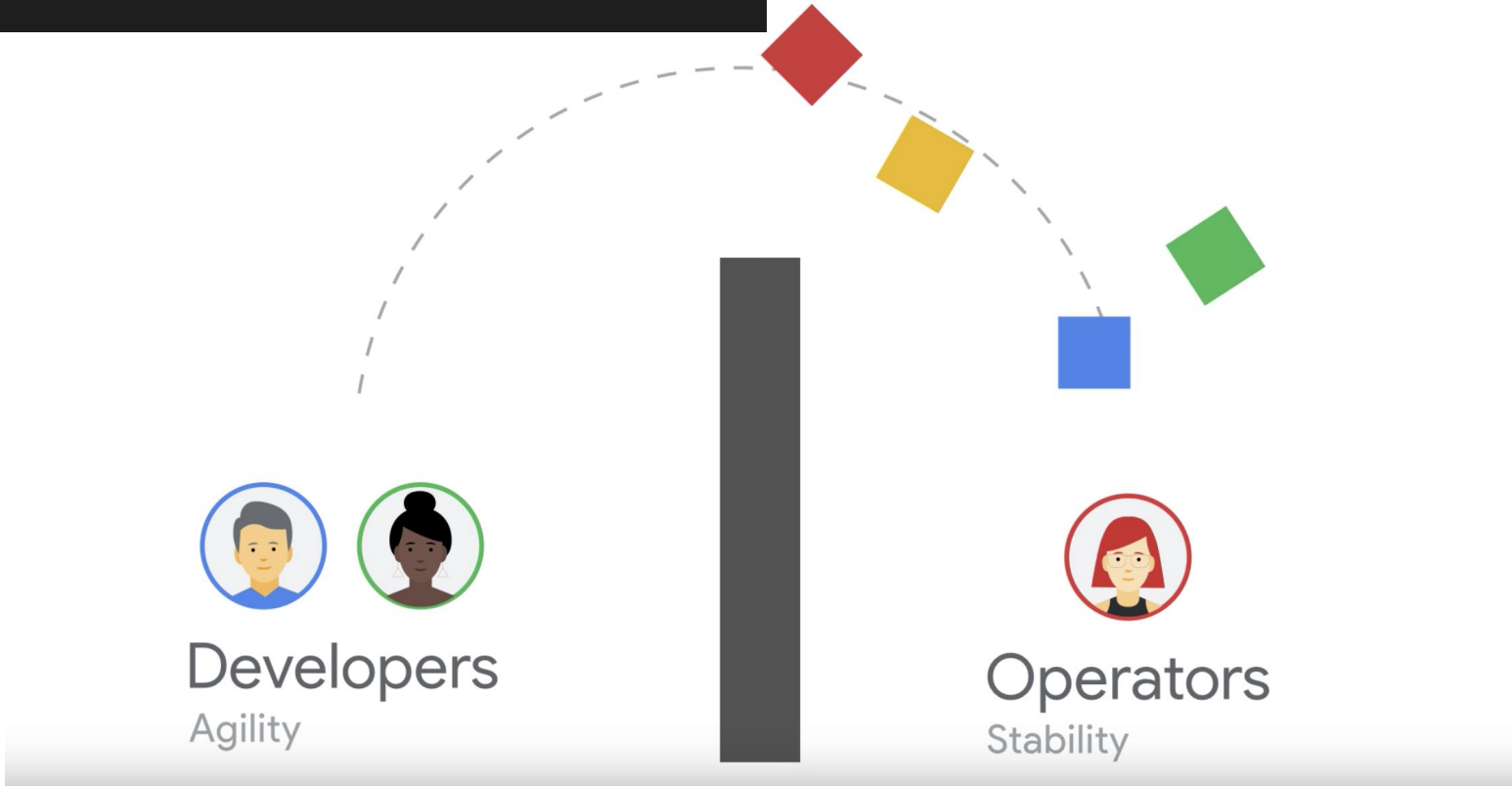
Type 7 – SRE Team (Google)



devopstopologies.com

https://www.infoq.com/articles/devops-team-topologies

class SRE implements DevOps



https://www.youtube.com/playlist?list=PLIivdWyY5sqJrKl7D2u-gmis8h9K66qoj
https://www.youtube.com/watch?v=uTEL8Ff1Zvk&list=PLIivdWyY5sqJrKl7D2u-gmis8h9K66qoj

Reduce
Organization Silos

降低組織穀倉效應

Accept
Failure as Normal

接受失敗是正常的

Implement
Gradual Change

漸進式改變

Leverage
Tooling & Automation

善用工具與自動化

Measure
Everything

量測任何數據

https://www.youtube.com/playlist?list=PLIivdWyY5sqJrKl7D2u-gmis8h9K66qoj
https://www.youtube.com/watch?v=uTEL8Ff1Zvk&list=PLIivdWyY5sqJrKl7D2u-gmis8h9K66qoj

共享擁有權
Share ownership

Reduce
Organization Silos

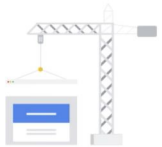
定義服務層級目標
SLOs & Blameless PMs

Accept
Failure as Normal

降低失敗成本
Reduce costs of failure

Implement
Gradual Change

將”苦工”自動化
Automate this year's job away



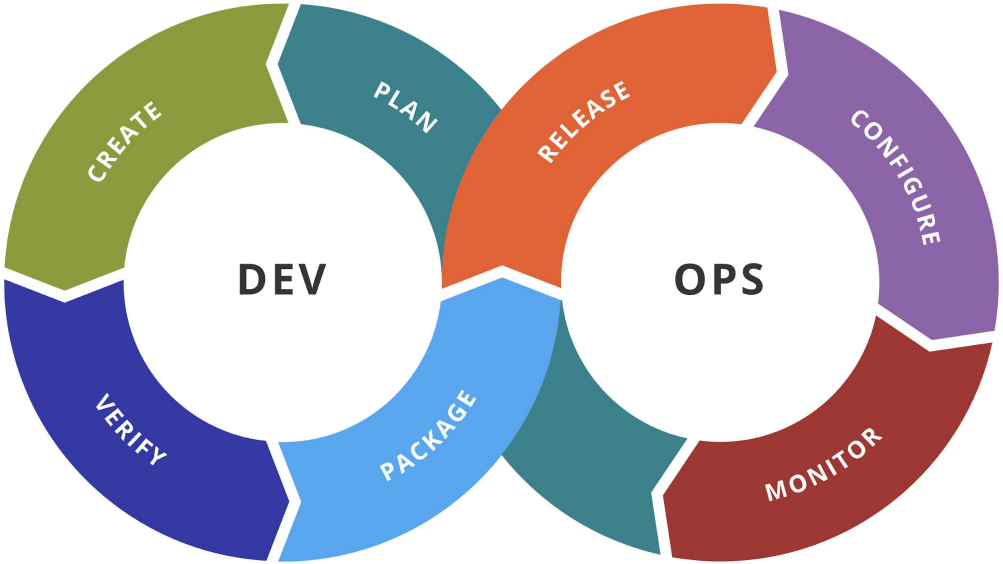
Leverage
Tooling & Automation

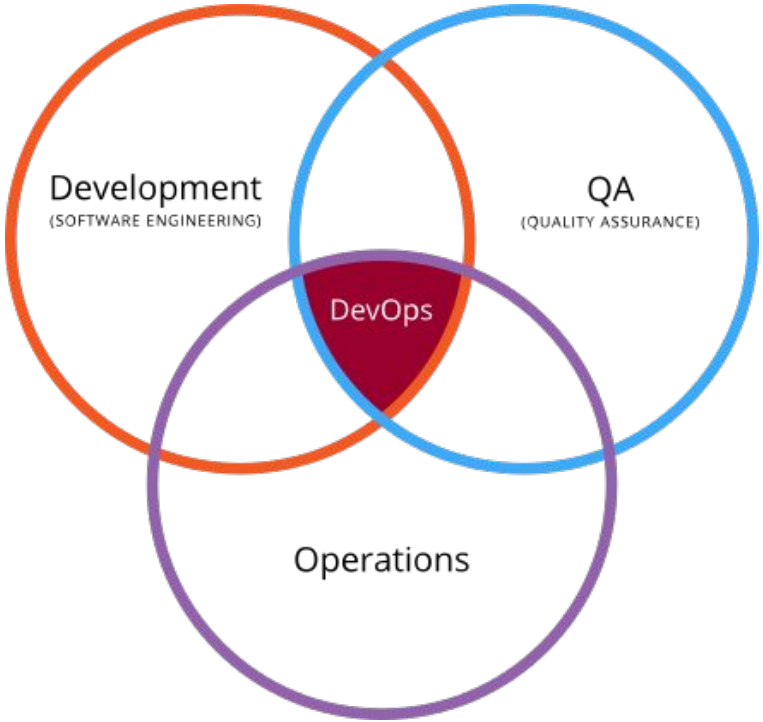
量測”苦工”和可靠度
Measure toil and reliability

Measure
Everything

https://www.youtube.com/playlist?list=PLIivdWyY5sqJrKl7D2u-gmis8h9K66qoj
https://www.youtube.com/watch?v=uTEL8Ff1Zvk&list=PLIivdWyY5sqJrKl7D2u-gmis8h9K66qoj

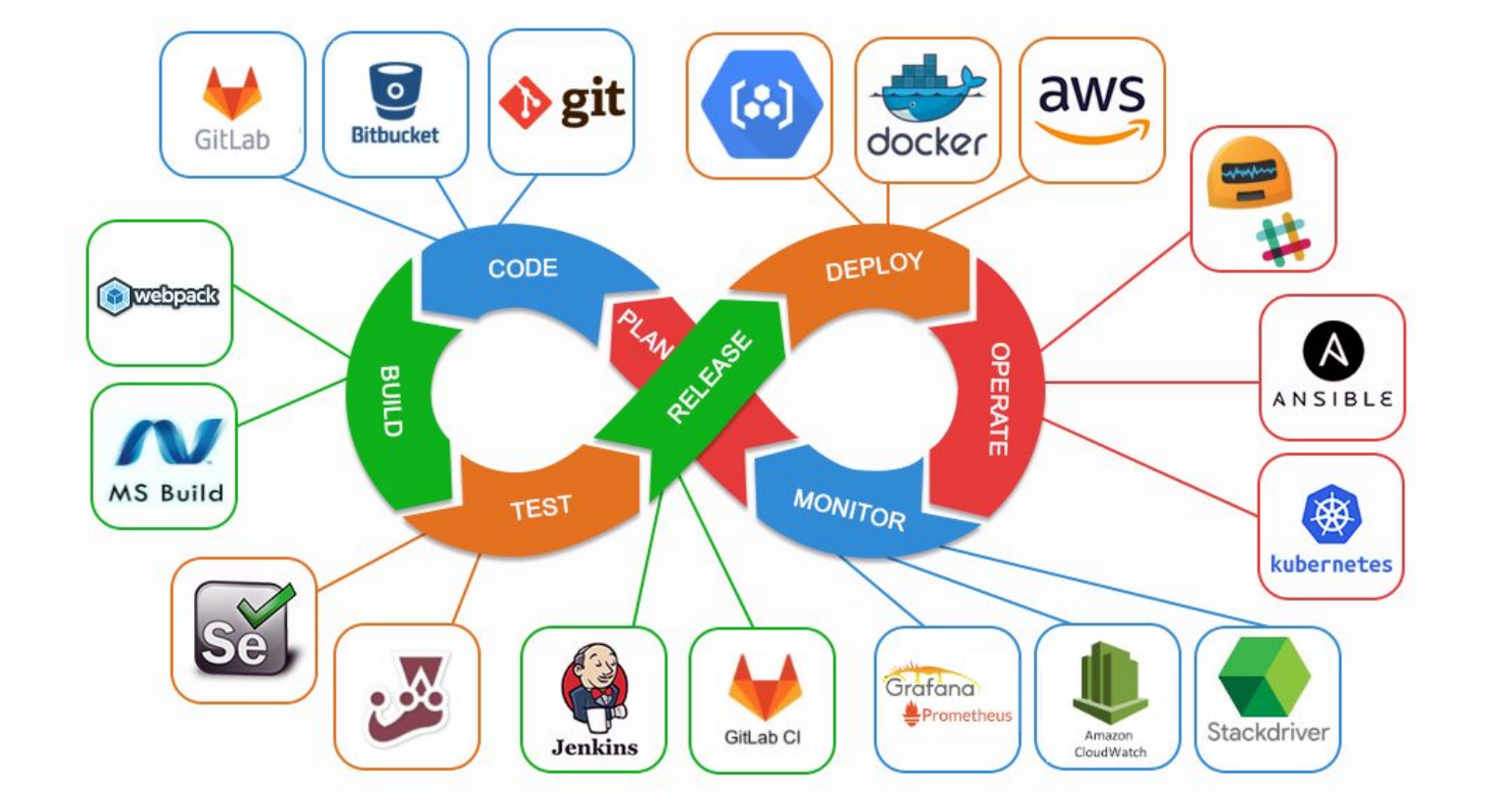
DevOps Cycle





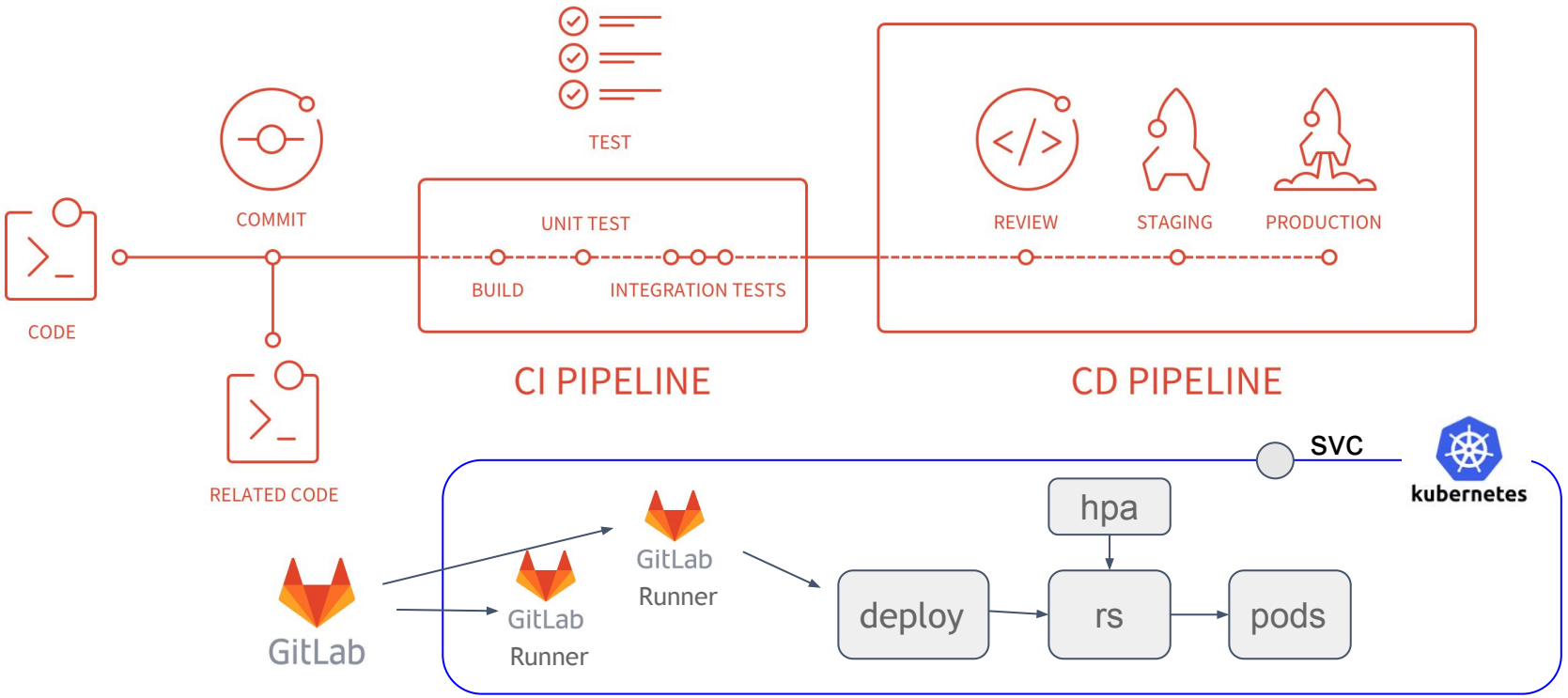
https://en.wikipedia.org/wiki/DevOps

DevOps 工具鏈


GitLab
Bitbucket
git
docker
aws
webpack
MS Build
ANSIBLE
kubernetes
Se
Jenkins
GitLab CI
Grafana
Prometheus
Amazon CloudWatch
Stackdriver
CODE
PLAN
BUILD
TEST
RELEASE
DEPLOY
OPERATE
MONITOR

部署流水線 Deployment Pipeline

https://docs.gitlab.com/ee/ci/



https://docs.gitlab.com/runner/install/kubernetes.html

91APP

耶！v1.6 開始有支援 RBAC了！
嗯，那有沒有比較簡單的方式，讓團隊容易監控？

Operate (Configure)
操作 (配置)

Plan → Code → Build → Test → Release →
Deploy → Operate → Monitor → Plan ...

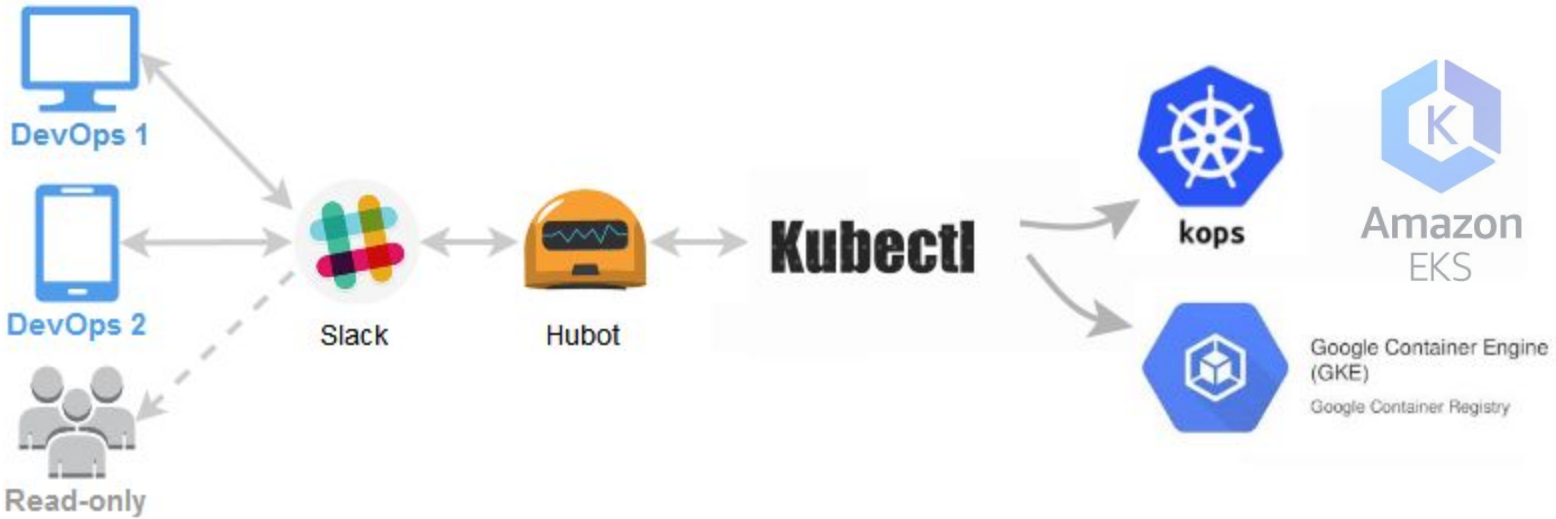
BB-8 專案

圖片來源：https://www.sideshowtoy.com/wp-content/uploads/2017/09/star-wars-the-last-jedi-bb-9e-sixth-scale-hot-toys-silo-903188.png

目的

- 資訊透明
- 合適的權限
- 隨時隨地只要有網路
- 降低權限管理複雜度

Slack + Hubot


DevOps 1
DevOps 2
Read-only
Slack
Hubot
Kubectl
kops
Amazon
EKS
Google Container Engine
(GKE)
Google Container Registry

其它可能方向

- canary deployment
- blue-green deployment switch
- filter exception log
- slack message button
- shopify chatOps - Incident Management

Timothy Liu [RD] 劉峰全 2:40 PM
@bb-8 top pod

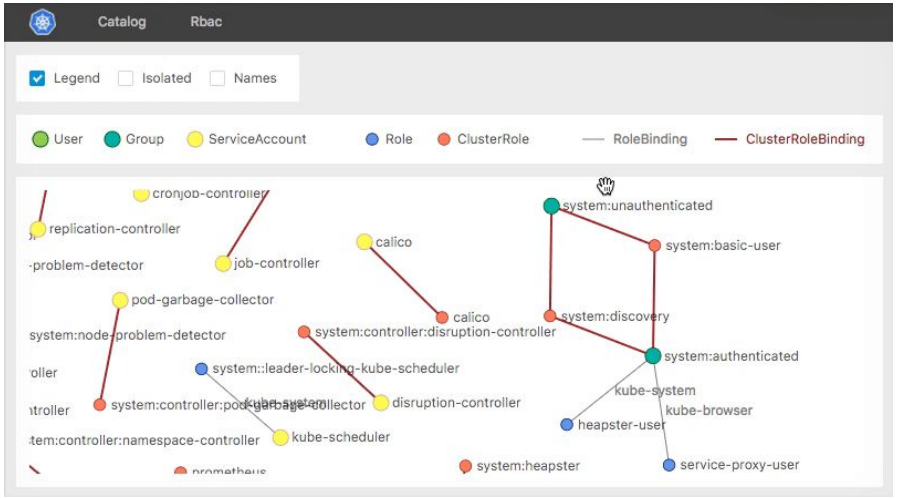
bb-8 APP 2:40 PM
再等一下！, 幫你查一下 prod 環境的 pod 資訊

bb-8 APP 2:40 PM
added this Plain Text snippet: timothyliu 要的 prod 環境的 pod 資訊

```
NAME                                                  CPU(cores)
MEMORY(bytes)
catalog-service-55df6c7db9-7l4rv                      1m           80Mi
catalog-service-55df6c7db9-bcflk                      1m           72Mi
catalog-service-55df6c7db9-666lc                      1m           95Mi
prod-cache2-redis-ha-server-6675796547-cn42c          1m           18Mi
catalog-service-new-779b789857-zwttq                  0m           79Mi
prod-cache-redis-2367068076-4xxxk                     1m           11Mi
catalog-service-55df6c7db9-jdttx                      2m           79Mi
catalog-service-55df6c7db9-s6swc                      2m           81Mi
prod-cache2-redis-ha-server-6675796547-nrnbg          1m           16Mi
catalog-service-55df6c7db9-cm49s                      1m           89Mi
prod-cache2-redis-ha-sentinel-59778c5cc-59pft         2m           5Mi
prod-cache2-redis-ha-server-6675796547-h6vf9          1m           16Mi
catalog-service-55df6c7db9-4g6dm                      1m           111Mi
catalog-service-55df6c7db9-s7224                      1m           98Mi
letterpress-service-67f6dccc56-5qgsq                  0m           87Mi
catalog-service-55df6c7db9-sk58r                      1m           89Mi
prod-cache2-redis-ha-sentinel-59778c5cc-fdsfd         2m           5Mi
letterpress-service-67f6dccc56-bwztb                  0m           83Mi
prod-cache2-redis-ha-sentinel-59778c5cc-9gxdf         3m           5Mi
letterpress-service-67f6dccc56-bwcqq                  0m           99Mi
catalog-service-55df6c7db9-25qch                      1m           78Mi
redis-commander-57d5b76fcd-pdr6v                      4m           55Mi
```

RBAC 輔助工具

- audit2rbac - Autogenerate RBAC policies based on Kubernetes audit logs
- kubernator - 低階版的 K8S Dashboard



91APP

如何設定資源的需求和限制？

Plan & Test
計劃和測試

Plan → Code → Build → Test → Release →
Deploy → Operate → Monitor → Plan ...

如何定義資源的需求和限制？

91APP

deployment 中的 .spec.temlate.spec.containers[].resources

```
resources:
  limits:          # 省略則可能用完 node 的 cpu/memory
    cpu: 600m      # 容器每 100ms 可用的 CPU 時間總量
    memory: 400Mi  # 超過，則 pod 將會被終止
  requests:        # 省略則同 limits
    cpu: 300m      # 影響 hpa 擴展條件
    memory: 200Mi  # 超過，則可能被驅逐(資源不足)
```

兩者都省略 → 常常吃到飽 → 放棄治療

如何定義資源的需求和限制？

- 觀察一般使用情形
  - `$ kubectl top pod`
  - k8s dashboard
  - prometheus + grafana
- 透過壓測／負載測試找出關連性：

  cpu/memory vs. latency or 錯誤率 (系統 KPI)

locust (蝗蟲)

91APP

- **Locust** is an easy-to-use, distributed, user load testing tool. Intended for load testing web sites (or other systems) and figuring out how many concurrent users a system can handle.
- http://locust.io/
- Python (native), Golang, Java

安裝 locust (蝗蟲)

- **helm install** --name loadtest \
  -f values.yaml **stable/locust**
- 測試程式放進 configmap

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: loadtest-worker
data:
  tasks.py: |
    from locust import HttpLocust, TaskSet, task
    import json, requests
    h = {
        "Host": "theme1.91dev.tw",
        "User-Agent":"locust",
    }
    class WebsiteTasks(TaskSet):
        def on_start(self):        # prewarm
            self.client.get("/", headers=h)
        @task(100)
        def index(self):
            self.client.get("/", headers=h)
        @task(5)
        def healthcheck(self):
            self.client.get("/health")
...
```

Load Test - locust (蝗蟲)

91APP

```
master:
  config:
    target-host: http://myservice.prod.svc.cluster.local:3005
…
worker:
    config: # all files from tasks folder are mounted under `/locust-tasks`
      locust-script: "/locust-tasks/tasks.py"
    replicaCount: 5
    resources:
      limits:
        cpu: 100m ...
```

Load Test

- 可以只**壓測程式本身**，不包含其它網路、LB等架構
- 參考**歷史最高 loading** 相關數據 (rps, resp time, ...)
- 壓測出**單一 pod** 在不同的 requests/limits x cpu/memory 之 reponse time/error rate，以求出較好的設定
- 調整 **hpa** 中合適的 targetCPUUtilizationPercentage 值或 Custom Metric (requests, ...)

Stress Test

- 設定值可能隨著功能的增減而有所不同，可考慮定期自動壓測
- 考慮使用 node selector/taint 將 測試端和被測端分開不同 node

LOCUST
A MODERN LOAD TESTING TOOL

STATUS RUNNING 10 users Edit | RPS 7.7 | FAILURES 0% | STOP | Reset Stats

Statistics Failures Exceptions Download Data

Type	Name	# requests	# fails	Median	Average	Min	Max	Content Size	# reqs/sec
GET	GetFeature_RoadSchedule	37	0	300	303	265	375	700	0.1
GET	GetLegendGraphic_RoadSchedule	55	0	46	48	30	65	605	0.1
GET	GetMap_RoadSchedule	1181	0	94	102	62	342	334	7.5
	Total	1273	0	94	105	30	375	356	7.7

https://www.slideshare.net/CompassInformatics/stress-testing-and-analysing-mapserver-performance

Stress Test

91APP

- 視覺化報表 - cloudwatch
  https://www.concurrencylabs.com/blog/how-to-export-locust-metrics-to-cloudwatch/
- 視覺化報表 - bokeh
  https://steelkiwi.com/blog/load-testing-python-locust-testing-and-bokeh-vis/
- 可透過 CLI 控制 master
- How to Run 50,000 Concurrent Users from Multi GEOs Using Locust.IO (SaaS)

91APP

如何跨微服務查問題？
如何監控那麼多服務？

Monitor
監控

Plan → Code → Build → Test → Release →
Deploy → Operate → Monitor → Plan ...

每一個 request 唯一識別碼

91APP

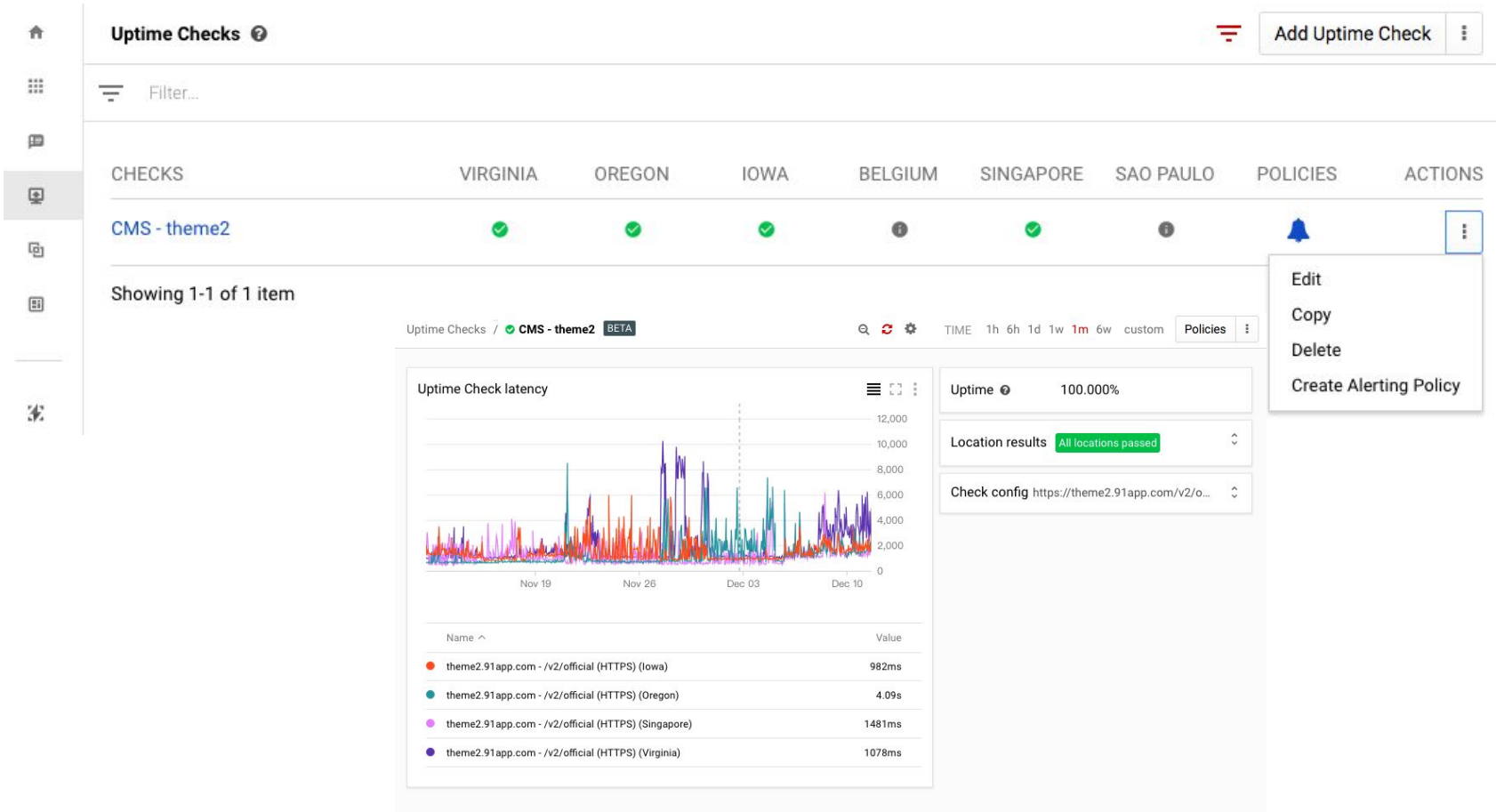
- NGINX
  - $request_id (http header)
    unique request identifier generated from 16 random bytes, in hexadecimal (1.11.0)
- aws Cloudfront
  - X-Amz-Cf-Id (http header)
  - x-edge-request-id (access log)
- …

APM 應用程式效能管理

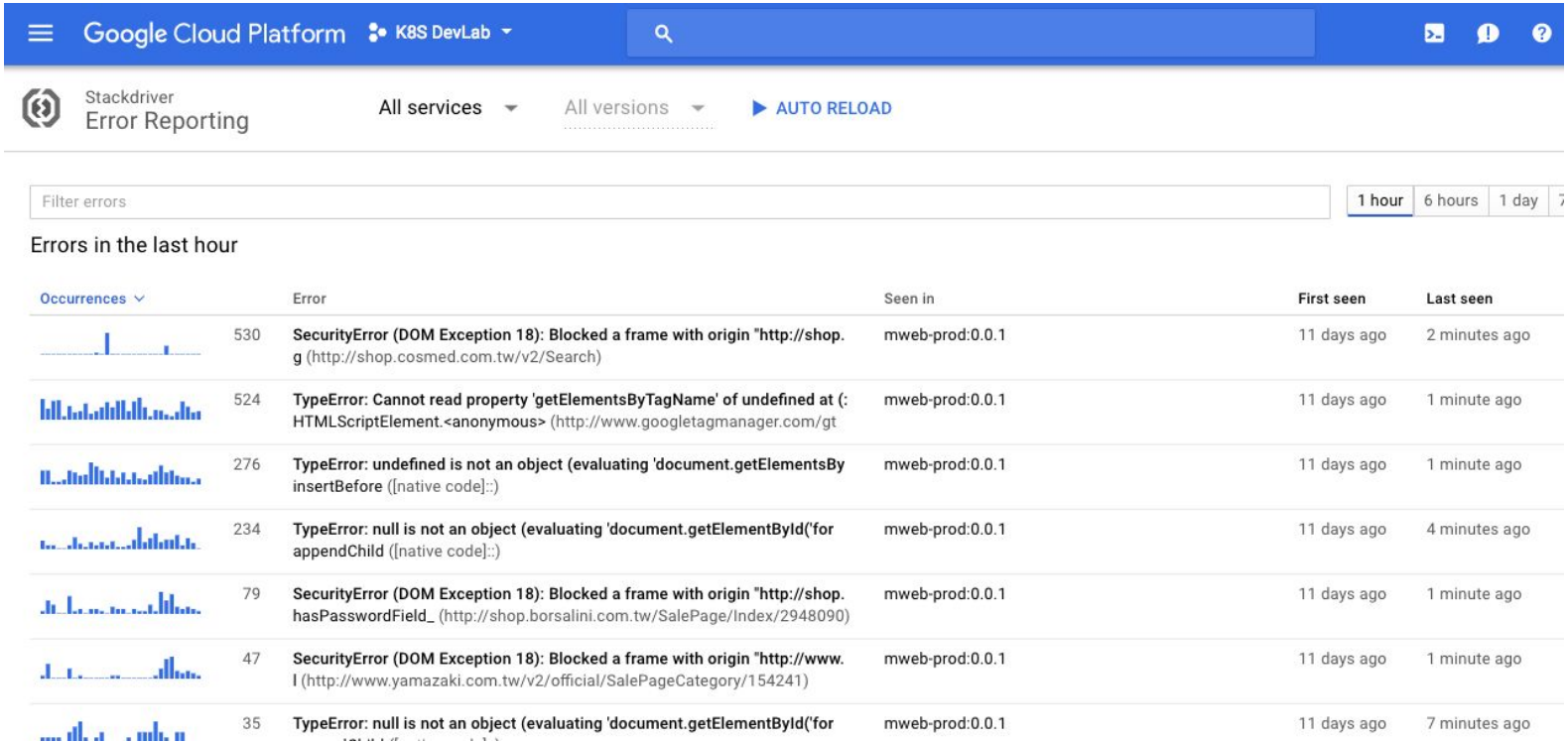
91APP

- Stackdriver APM
- Zipkin (open distributed tracing framework, 基於 Google Dapper 論文)
- Azure Application Insights
- aws x-ray
- ...

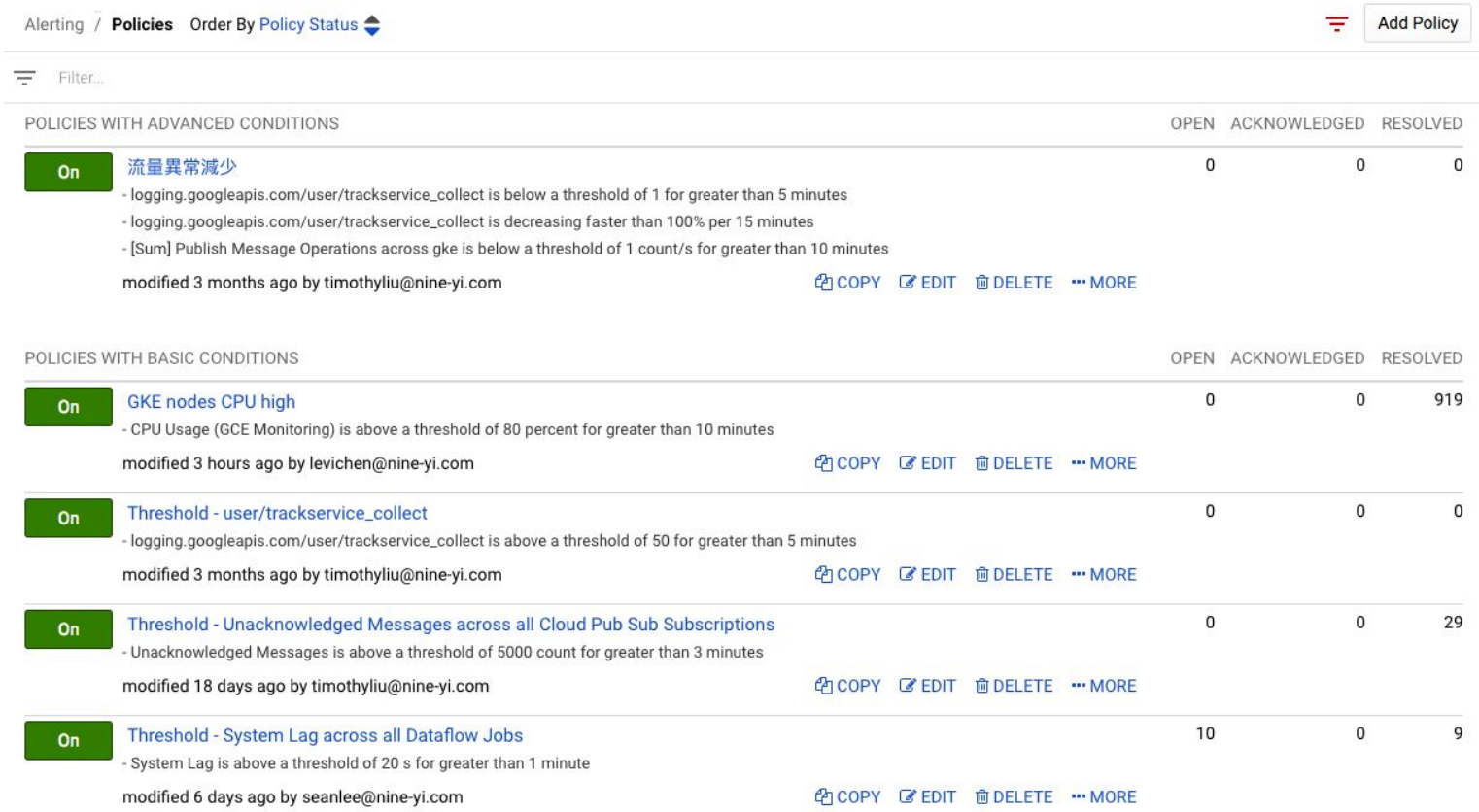
Uptime Checks (Stackdriver)



Error Reporting (Stackdriver)



Alerting Policies (Stackdriver)



System Metric & Log Metric

91APP

- Log Metric
  - Application Level Metric
    - start/stop
    - latency detail
    - ...
  - Business Level Metric
    - transaction
    - online users
    - ...

SRE Golden Signals

91APP

- How to Monitor the SRE Golden Signals
  - **Rate** —Request rate, in requests/sec
  - **Errors** —Error rate, in errors/sec. (5xx, 4xx, ...)
  - **Latency** —回應時間, including queue/wait time, in milliseconds. 通常要看 LB 的 metric.
  - **Saturation** — 系統超載, 例如 queue depth, `kubectl get hpa`, `kubectl top pod`
  - **Utilization** — 系統使用率, 例如 `kubectl top node`

91APP

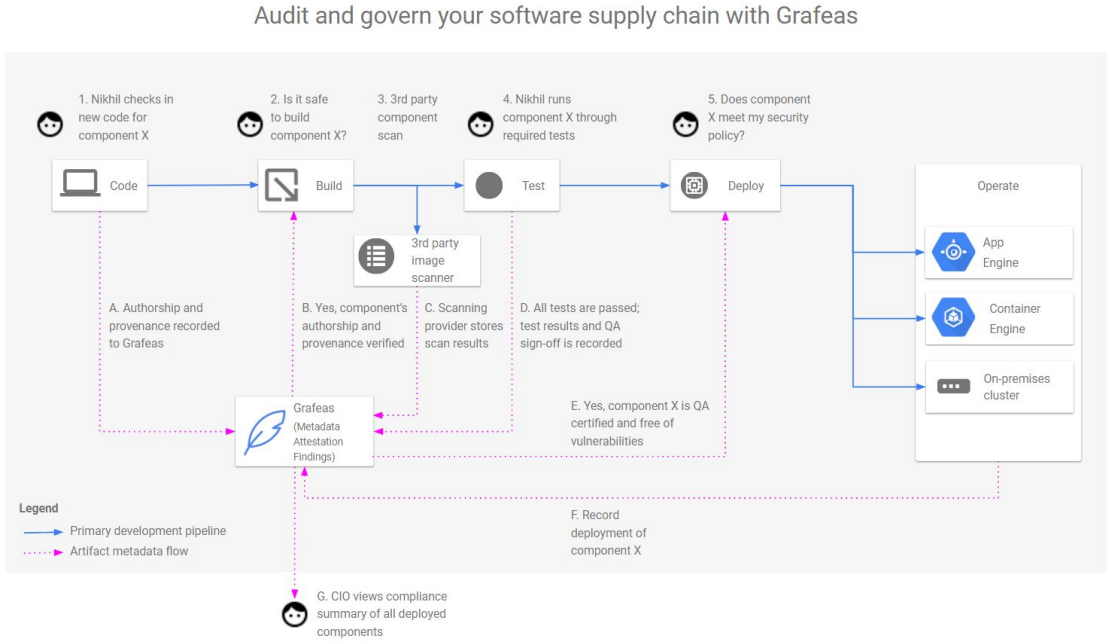
我的服務如何更安全？

如何將資安左移？

Plan → Code → Build → Test → Release →
Deploy → Operate → Monitor → Plan ...

Security

- Grafeas
  開源容器安全工具 - 用於收集和彙總特定的元數據，為用戶提供了一個標準化的方式，審核和管理軟件供應鏈。
  ([tutorial](), [g! blog]())



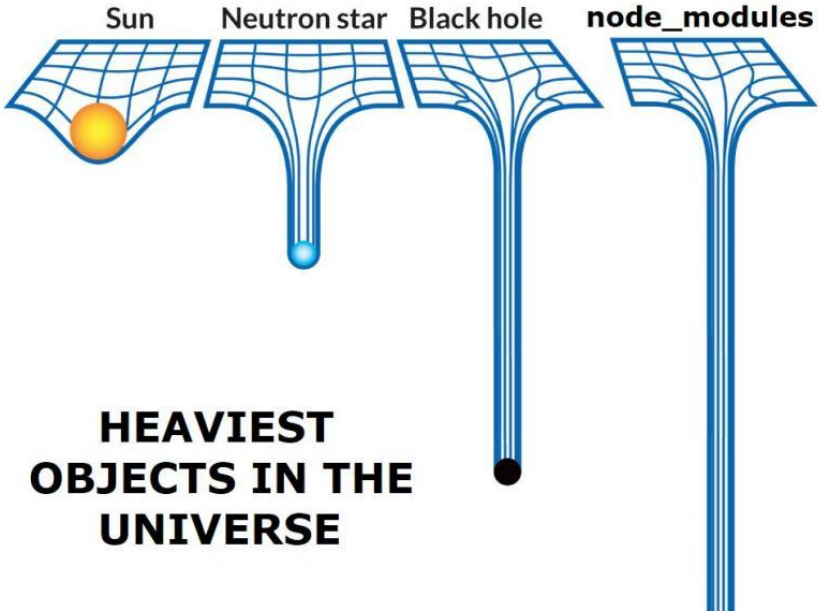
Security

91APP

- CIS Benchmark https://www.cisecurity.org/cis-benchmarks/
  - A set of scripts checks best-practices of Kubernetes installations https://github.com/neuvector/kubernetes-cis-benchmark
- 最佳實踐 https://github.com/freach/kubernetes-security-best-practice

Node 之父 Ryan Dahl

https://zhuanlan.zhihu.com/p/37637923



http://tinyclouds.org/jsconf2018.pdf

node.js security

91APP

- npm audit

  $ npm audit --registry https://registry.npmjs.org/

```
                    === npm audit security report ===

# Run  npm install body-parser@1.18.3  to resolve 1 vulnerability
┌───────────────┬──────────────────────────────────────────────┐
│ Low           │ Regular Expression Denial of Service         │
├───────────────┼──────────────────────────────────────────────┤
│ Package       │ debug                                        │
├───────────────┼──────────────────────────────────────────────┤
│ Dependency of │ body-parser                                  │
├───────────────┼──────────────────────────────────────────────┤
│ Path          │ body-parser > debug                          │
├───────────────┼──────────────────────────────────────────────┤
│ More info     │ https://nodesecurity.io/advisories/534       │
└───────────────┴──────────────────────────────────────────────┘

# Run  npm install morgan@1.9.0  to resolve 1 vulnerability
┌───────────────┬──────────────────────────────────────────────┐
│ Low           │ Regular Expression Denial of Service         │
├───────────────┼──────────────────────────────────────────────┤
│ Package       │ debug                                        │
├───────────────┼──────────────────────────────────────────────┤
│ Dependency of │ morgan                                       │
├───────────────┼──────────────────────────────────────────────┤
│ Path          │ morgan > debug                               │
├───────────────┼──────────────────────────────────────────────┤
│ More info     │ https://nodesecurity.io/advisories/534       │
└───────────────┴──────────────────────────────────────────────┘
```

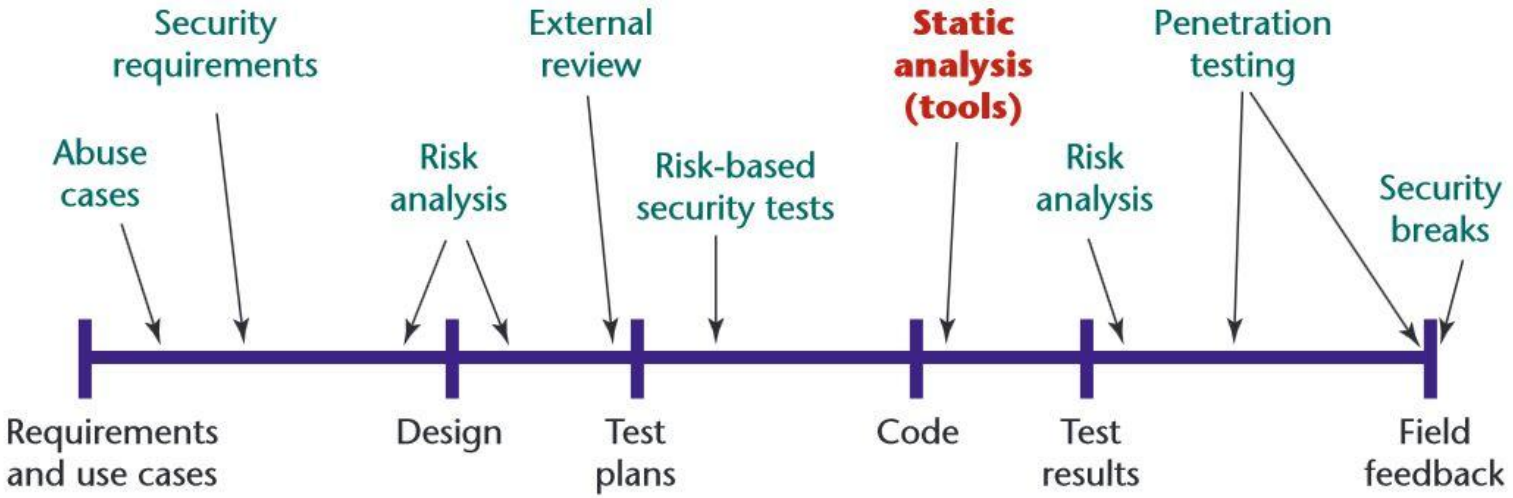
security headers

91APP

- security headers
  透過伺服器回應的 header 讓瀏覽器依此保護使用者
  https://securityheaders.com/
  - Strict-Transport-Security
    X-Frame-Options
    X-XSS-Protection
    X-Content-Type-Options
    Content-Security-Policy
    Referrer-Policy

static analysis 靜態分析

- https://github.com/mre/awesome-static-analysis
- Software Life Cycle



https://www.synopsys.com/blogs/software-security/software-security/

91APP

如何降低 K8S 昇級的風險？

Operate
操作

Plan → Code → Build → Test → Release →
Deploy → Operate → Monitor → Plan ...

kops 昇級失敗事件

日期: 2017-11-xx

目標: 正式環境 k8s cluster (kops), 從 v1.7.x -> v1.8.3

現象: 昇級後的 node 無法正常加入或放入 pod

解法: [快解] 中斷昇級, 透過 kops edit 執行改回原值後再 update

[正解?] kubelet 啟始參數補缺 -
--runtime-cgroups=/systemd/system.slice ...

kops 昇級失敗事件

日期: 2018-05-03

時間: 晚上10點

目標: 測試環境 k8s cluster (kops), 從 v1.8.6 -> v1.9.3

現象: all 3 masters 無法連入

解法: [快解] 中斷昇級, 透過 kops edit 執行改回原值後再 update

[正解?] debian AMI → ubuntu AMI

Upgrade like blue-green

91APP

- 感謝 iKala 分享了他們在 GKE 的安全做法
- https://cloudplatform.googleblog.com/2018/06/Kubernetes-best-practices-upgrading-your-clusters-with-zero-downtime.html
- create node pool 2
  → cordon pool 1 → drain pool 1
  → upgrade pool 1 → uncordon pool 1
  → cordon/drain/delete pool 2 ...

其它

Job/CronJob

- cron table 格式定義
- create cronjob -> job -> pod
- running pod 不受 job/cronjob 變更的影響
- 可控制平行處理的 job 數
- 量大時 (包含歷史) 可能對 etcd 產生壓力 (2000 pods), `kubectl get pod` 跑不出來

關於 Container image tags 用法建議

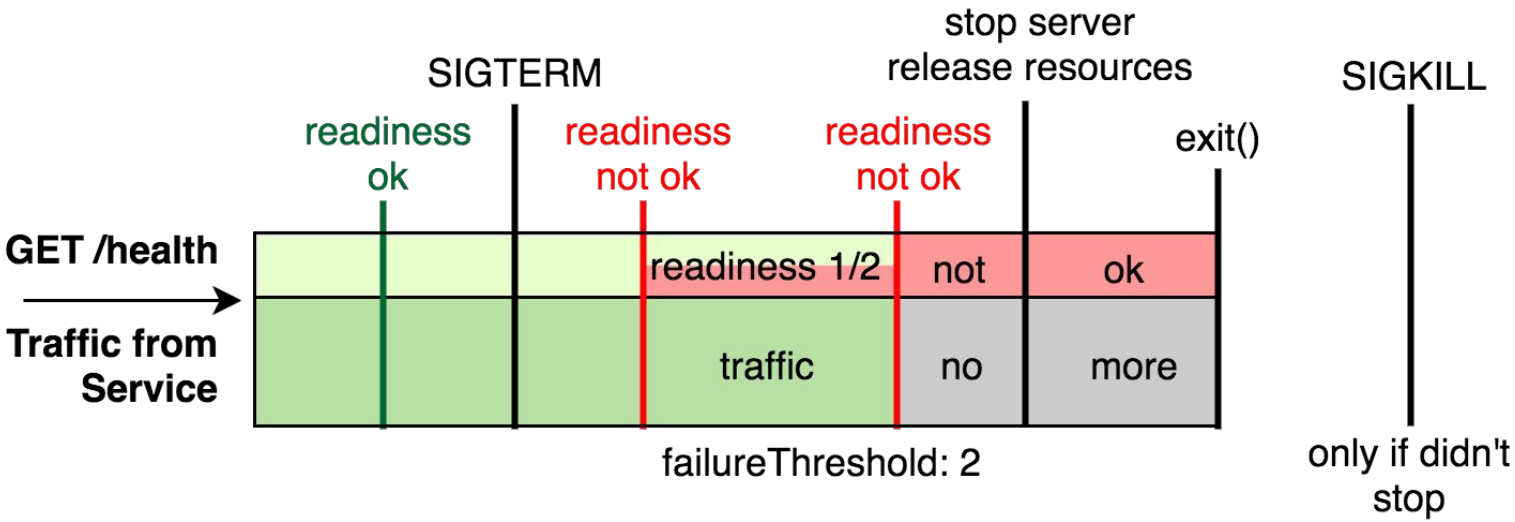
91APP

- 自已的 image 不建議用 :latest
- 須搭配 imagePullPolicy: always ⇒ 否則無法預期結果
- 若 stage 和 prod 用同一個 registry 則無法區別
- 當大量 autoscale out 時，可能用盡網路頻寬

Graceful shutdown

91APP

- https://blog.risingstack.com/graceful-shutdown-node-js-kubernetes/ (github)



Graceful shutdown - 正確結束資源

91APP

```
process.on('SIGTERM', function onSigterm () {
  console.info('Got SIGTERM. Graceful shutdown start', new Date().toISOString())
  // start graceful shutdown here
  shutdown()
})
```

Graceful shutdown - 降低問題成本

91APP

```
if (err.toString().indexOf('Deadline Exceeded') > -1) {
  global.isReadness = false
  setTimeout(() => {
    process.exit()
  }, process.env.PROCESS_KILL_TIME)
  logging.error('Process exit for PubSub Deadline Exceeded')
}
```

Graceful shutdown - 降低問題成本

91APP

- ```
  app.get('/_ah/health/online', (req, res) => {
    if (global.isReadness) {
      res.status(200).send('ok')
    } else {
      res.status(503).send('Pod offline')
    }
  })
  ```
  ```

關於 GKE 的經驗

91APP

- 如何避免重建 GKE cluster (追加)
  - Creating VPC-native clusters using Alias IPs
    - Pod IP 在本地可在 GCP 網路內路由(包括通過 VPC network peering),不再使用路由配額。
    - Pod IP 在網路中預先保留，防止與其他計算資源衝突。
    - 網絡層可以執行反欺騙檢查以確保出口流量不會與任意源IP 一起發送。雲端路由器可以通過BGP 發布別名 IP 。
    - Pod 的防火牆控件可以與其節點分開應用。
    - 別名 IP 允許 Pod 在不使用 NAT Gateway 的情況下直接訪問託管服務。

關於 GKE 的經驗 - 修正

91APP

- 跨不同”專案”的 GKE Cluster ~~無法~~建立內部連線
  - https://cloud.google.com/compute/docs/vpc/vpc-peering

    Container Engine with VPC Network Peering

    ~~Container Engine is **not supported** with VPC Network Peering. Container Engine containers can't communicate with VMs or workloads in peered networks~~
  - https://cloud.google.com/compute/docs/shared-vpc/provisioning-shared-vpc

    Shared VPC limitations

    ~~GKE clusters in a service project associated with an shared VPC network are **not supported**~~.

Awesome Kubernetes

91APP

- Best practice @ aws
  https://github.com/aws-samples/aws-workshop-for-kubernetes
- `kubectl get deploy -o json | jq -cr ".items[] | {name: .metadata.name, resources: .spec.template.spec.containers[].resources}"`
- http://kubernetesstatus.com/
- https://github.com/operator-framework
- https://www.openebs.io/

THANK YOU

如有任何建議或疑問，歡迎隨時與我們聯繫！