TO QUALITY, AND BEYOND

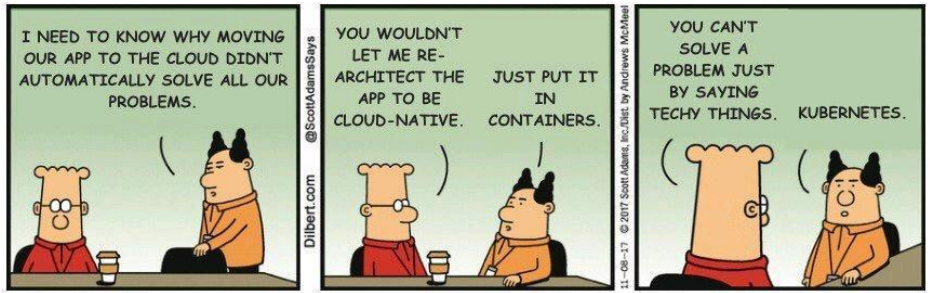
Brad Marshall, AARNet

Brad.marshall@aarnet.edu.au

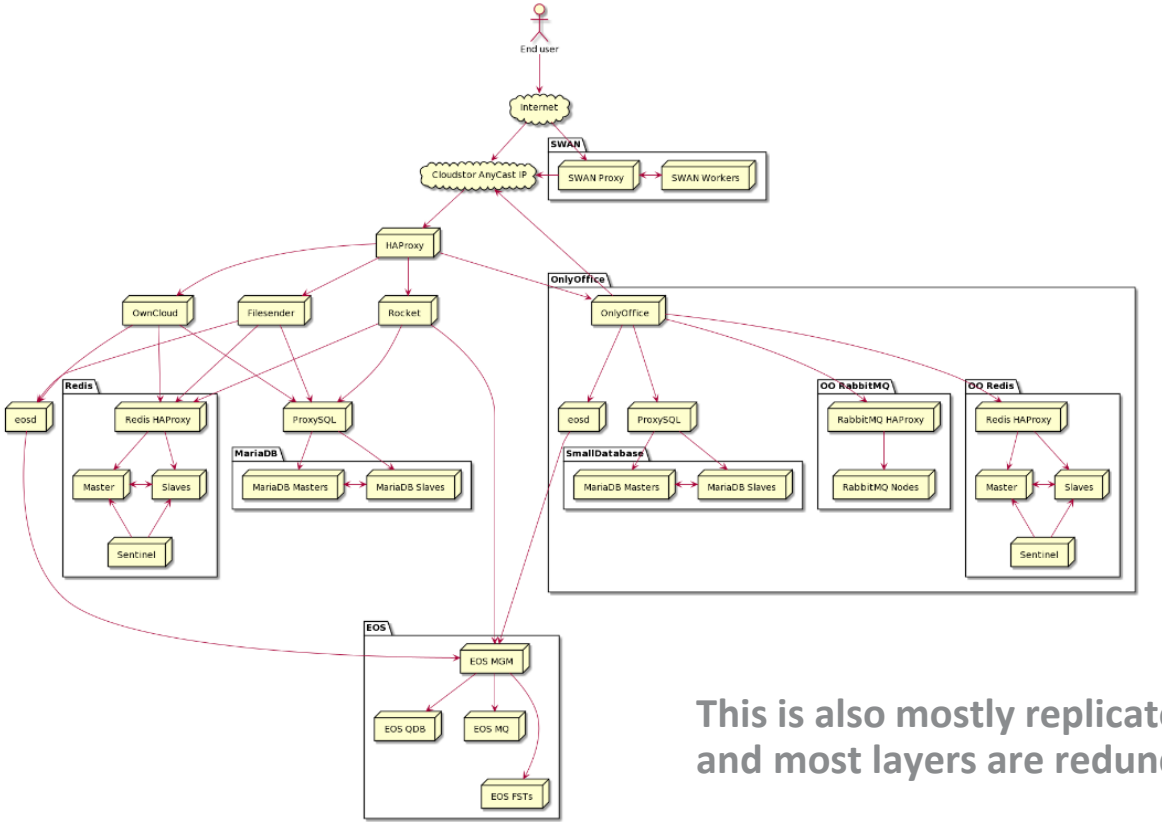
27 Jan 2021

WHAT IS CLOUDSTOR

- **Australia's largest cloud storage and sync website for research**
  - Owncloud with EOS backend
  - S3 shards using Minio and EOS
- **Four major sites – Brisbane, Melbourne, Canberra, Perth**
- **Number of users - ~ 100k users**
- **Number of files – 361M in Cloudstor Prime**
- **Growth rate – 215k files a day, 100T a month**



WHAT IS CLOUDSTOR REALLY



This is also mostly replicated per site, and most layers are redundant internally

QUALITY APPROACHES

- **The system is scaling rapidly, and seeing issues here and there**
- **Have had issues with users reporting performance issues at specific regions**
- **Found 3 main areas to start improving quality:**
- Scalable test environments
- Improved monitoring
- CI/CD

PLAN FOR TEST ENVIRONMENT

- **Reproducibility is the main problem here**
- **Current test environment is a single UAT cluster, makes testing multiple concurrent issues difficult**
- **Also conflating testing with development**
- Some development done locally, but doesn't reflect prod
- **So how can we fix it?**

QA ENVIRONMENT

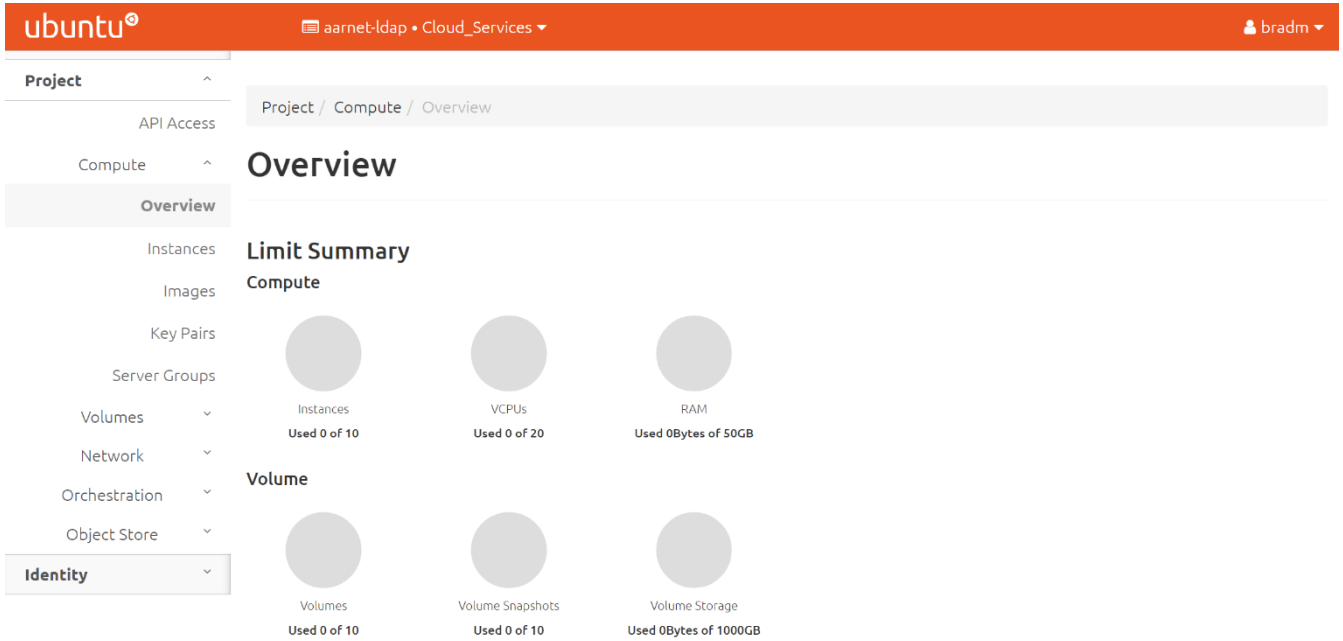
- New QA environment built based on Openstack using Canonical toolset including Ubuntu, MAAS (OS deployment, Juju (configuration management) and containers
- Terraform to spin up VMs
- Use same technology as prod to deploy (usually ansible or helm)
- Now we can have as many isolated environments as we need (within resource limits)
- Can build up complicated network to emulate production

QA MONITORING

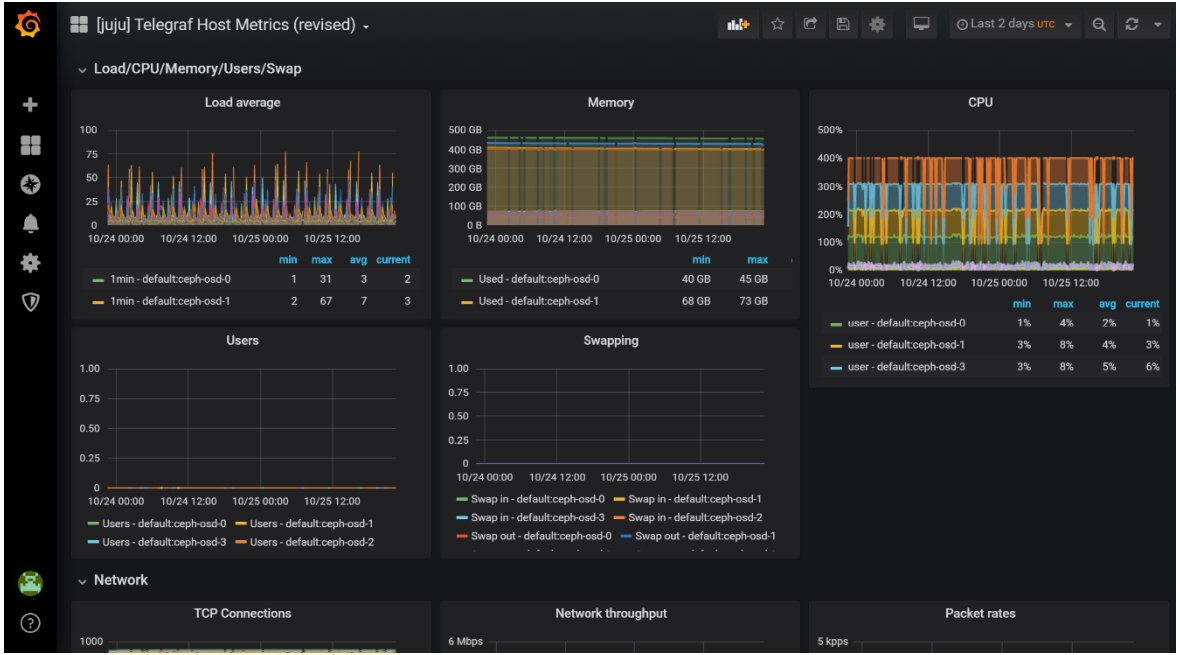
The QA environment has its own set of infra for monitoring and logging:

- **Nagios for monitoring**
- **Prometheus for metrics**
- **Elasticsearch for logging**
- **Grafana and graylog for visualisation**
- **Openstack Horizon for VM manipulation**
- **MAAS for OS installs**

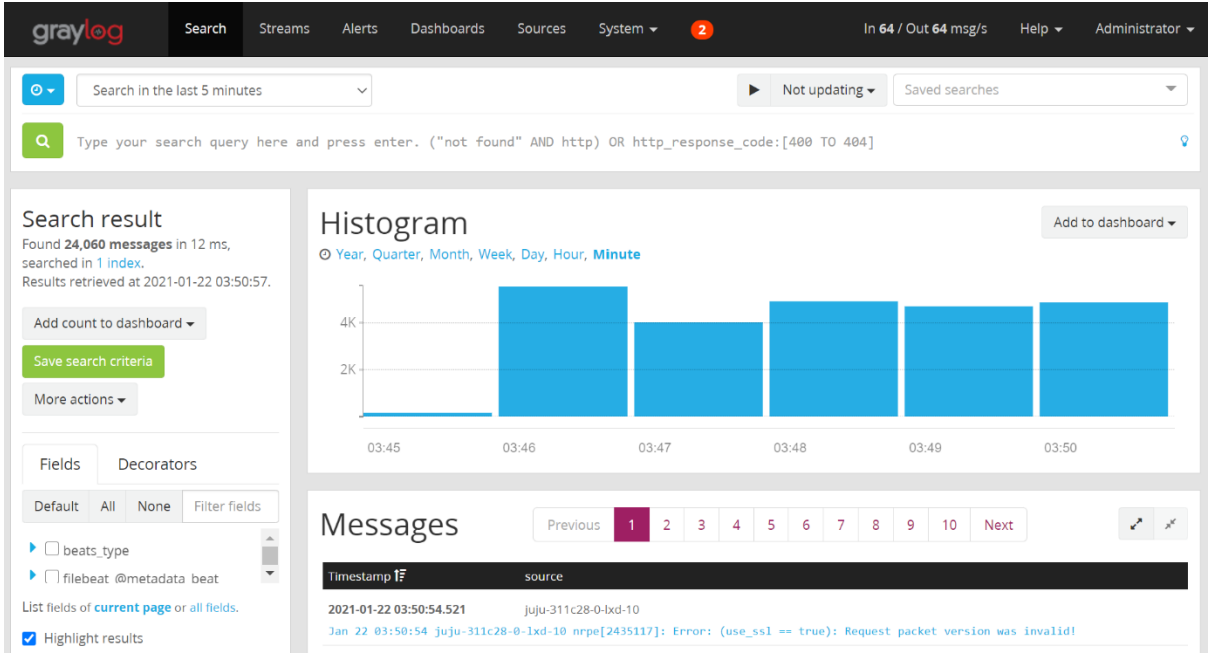
QA ENV DASHBOARD



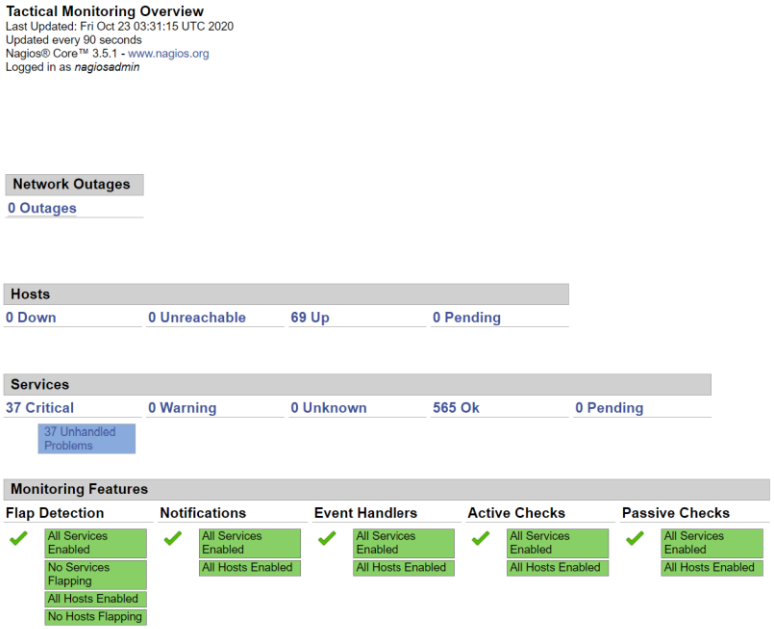
QA ENV GRAFANA



QA ENV LOGGING



QA ENV MONITORING



CURRENT DEPLOYMENT METHODOLOGY

- **Cobbler to deploy RHEL**
- PXE boot plus image deployment
- **Ansible to configure hosts**
- Configuration management tool
- **Mix of rancher 1.6 (essentially docker-compose) and rancher 2.2 (kubernetes) for containers**
- Container management
- **Kubernetes deploy via helm**
- Orchestration inside kubernetes

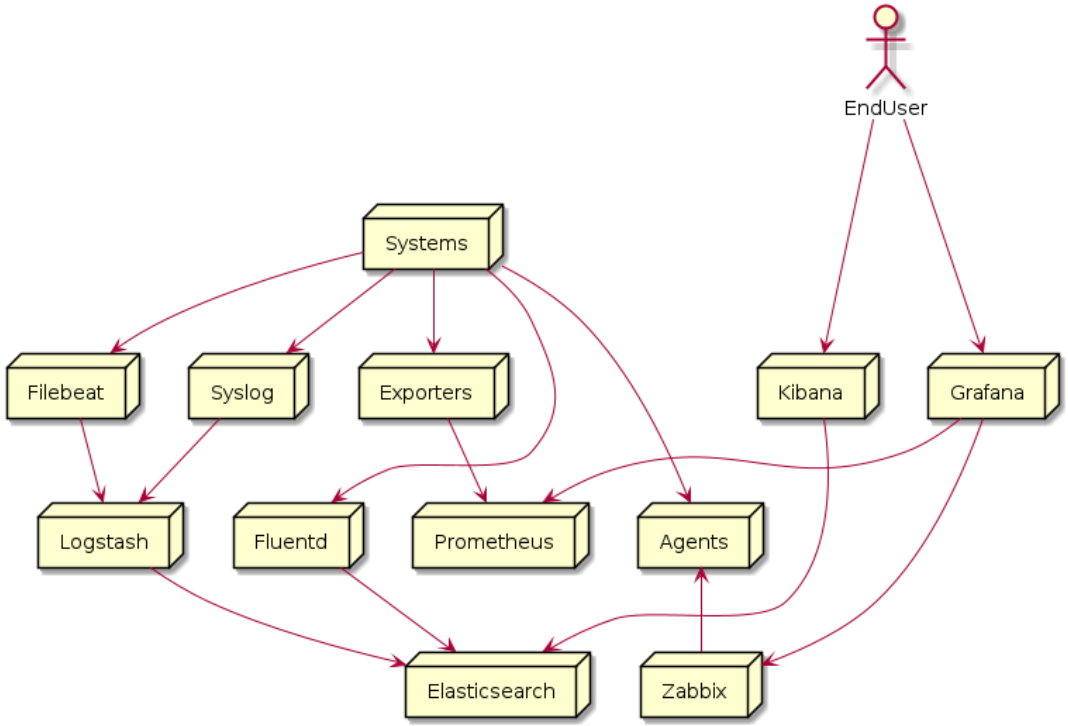
MONITORING PROBLEMS

- No end to end checks for performance metrics
- Limited visibility of whats happening inside the application
- Always more system metrics to be monitored

MONITORING

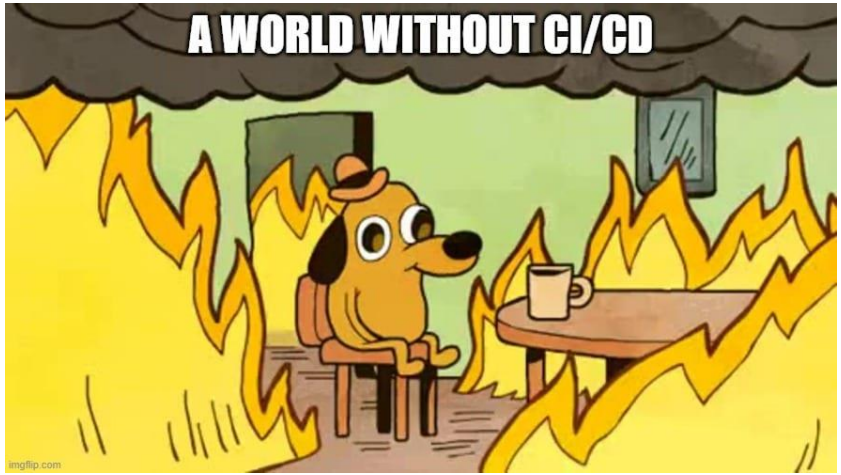
- Zabbix does most of the monitoring
- Grafana and Kibana for dashboarding and log searching
- Mostly doing system based metrics – ie, disk usage, cpu load, memory etc
- Starting to add Prometheus monitoring
  - Initially starting with node exporter (for system metrics) and cAdvisor (for container metrics)
  - Will evaluate AlertManager for sending alerts on these metrics
- Adding end to end performance metrics
  - Means no more surprises from customers

MONITORING ARCHITECTURE



CI/CD

- Using Jenkins for CI/CD
- Currently autobuilding and testing containers on git commit
- Doing some automated deployments to testing systems from git
- Allows us to ensure system is working after every change
- Removes manual steps from the process



CONCLUSION

- More visibility of what's going on means faster reaction times to problems
- Less user reporting issues that we don't know about, more proactive
- Hopefully develop feedback loops to improve systems
- Faster QA means faster deployments means less downtime
- Better tested services means better user experience

THANK YOU – ANY QUESTIONS?

BRAD.MARSHALL@AARNET.EDU.AU