Science & Technology Facilities Council

ATLAS jobs on Azure

Andrew Lahiff

STFC Rutherford Appleton Laboratory

22nd May 2017

Introduction

- RCUK Cloud Working Group Pilot Project
  - Aims to investigate
    - portability between on-premises resources & public clouds
    - portability between multiple public clouds
  - Using particle physics workflows as an example
    - initially CMS, but more recently ATLAS
- Provision virtual machines on demand?
  - this has been done for many years by many people
  - representatives from public clouds were more interested in containers (especially Google)
- Initially had credits from Amazon, Google & Microsoft
  - now only Azure Sponsorship left (expires 1st July)

Introduction

- Idea is to use Kubernetes as an abstraction layer
  - easily available on major public clouds
    - click a button, or
    - scripts designed to automate provisioning & management, or
    - via third-party company (e.g. StackPointCloud, KUBE2GO)
  - using *only* the Kubernetes API, can run work anywhere where Kubernetes is available
    - don't need to write code to deal with all the different cloud APIs
    - use only industry-standard open-source software
  - Kubernetes federations
    - a single API request can deploy across multiple Kubernetes clusters around the world

Running ATLAS jobs

- What do we deploy on Kubernetes?
  - squids
    - automatically replaced if they die or a node dies
    - accessed via a Kubernetes Service
    - provides stable VIP & round-robin load balancing
  - use the vacuum model to run HTCondor startds
    - custom controller
    - create as many startds as possible (up to resource limits)
    - backs off if they start failing or have no payload
  - proxy for authentication with ATLAS HTCondor pool
    - host cert stored as a Kubernetes Secret
    - a short-lived proxy is created regularly, provided to startds

Running ATLAS jobs

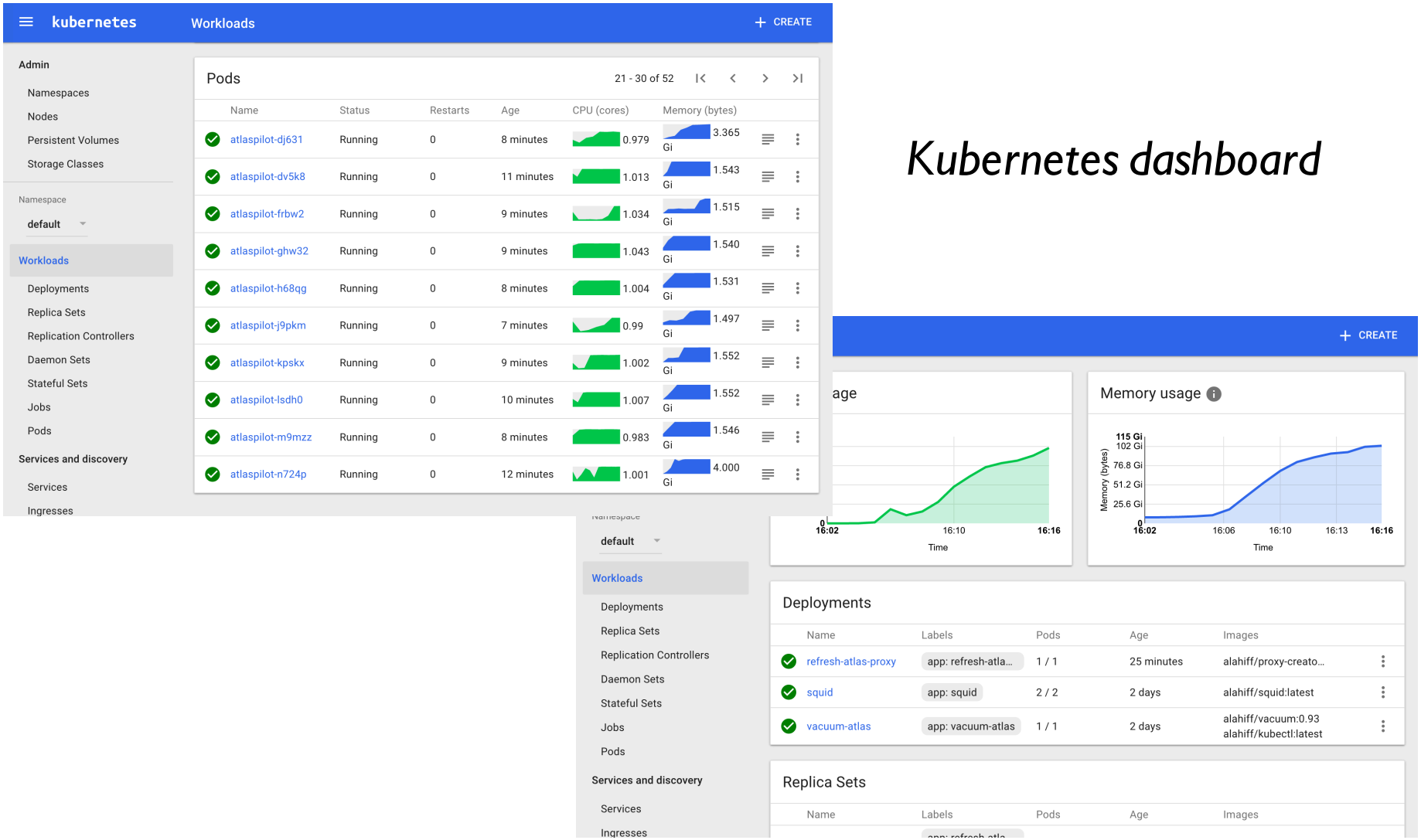
- Very simple to deploy
  - upload a certificate

```
kubectl create secret generic atlas-cert \
  --from-file=hostcert=hostcert.pem \
  --from-file=hostkey=hostkey.pem
```

  - update ATLAS site name in a json file
    - template used for creating ATLAS startds
  - deploy

```
kubectl create –f atlas.yaml
```

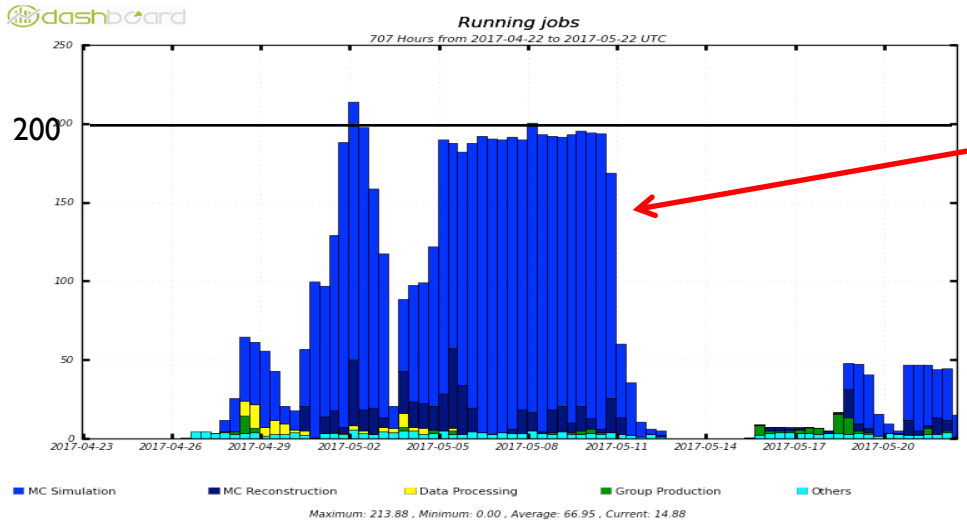
Running ATLAS jobs



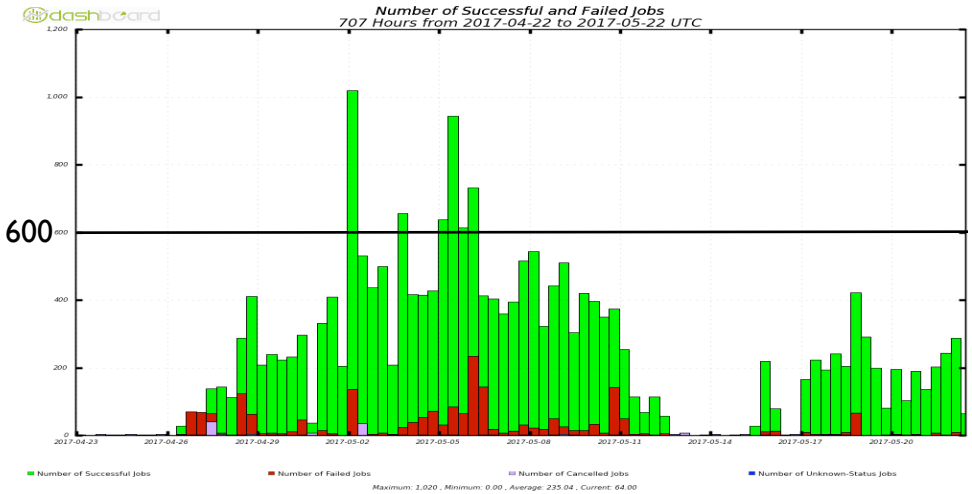
Kubernetes dashboard

Running ATLAS jobs

- Started running jobs on RAL-AZURE at the end of April



problem with aipanda135.cern.ch resulted in lack of jobs



http://bigpanda.cern.ch/site/RAL-AZURE_VAC/

Usage & cost monitoring

- Existing Azure billing info not really adequate
- Use Azure Python SDK to generate usage & cost data, send to InfluxDB

CVMFS

- CVMFS is still a complication
  - only option now is to use a single privileged container which
    - provides CVMFS
    - runs a HTCondor startd
  - not currently possible to have CVMFS & pilot in separate containers
    - different mount namespaces
  - installing CVMFS directly on Kubernetes nodes would break portability
  - in future hope to use Kubernetes itself to deploy CVMFS onto nodes in the cluster
    - mount propagation (e.g. PR #41683)

Issues encountered

- Connection between startds on Azure & external shadows are lost after a few minutes
  - Azure kills idle tcp connections after 4 mins
  - Reduce TCP_KEEPALIVE_INTERVAL from the default 600s
- By default emptyDirs are on the OS disk
  - OS disk is small (30GB), not local to VM
    - not enough space for job sandboxes, CVMFS cache, ...
    - local SSD unused!
  - current workaround:
    - use the Azure Container Service engine from github
    - modified to put /var/lib/kubelet on local disk
  - hopefully Microsoft will make this the default in future

Issues encountered

- Azure Container Service
  - doesn't yet support node autoscaling
    - Google Container Engine does
  - can only manually increase/decrease size of cluster
- Occasional strange kernel errors
  - still needs investigation

```
SLUB: Unable to allocate memory on node -1 (gfp=0x2080020)
  cache:
   ext4_extent_status(4409:0ce0c80c8cfe0e093b4b6c7a2e133616
   7fa357d72bb76b141591417ce5ba052a), object size: 40,
   buffer size: 40, default order: 0, min order: 0
  node 0: slabs: 79, objs: 8058, free: 0
```

Summary

- Kubernetes is a promising way of running LHC workloads
  - run jobs in exactly the same way on-premises or on public clouds
  - not quite ready for widespread production use
    - e.g. waiting for Kubernetes to support configurable mount propagation
- Future work
  - jobs staging out to Azure Blob storage
  - suggestions welcome!

Thanks to Microsoft for an Azure Research Award, Google & Amazon for credits