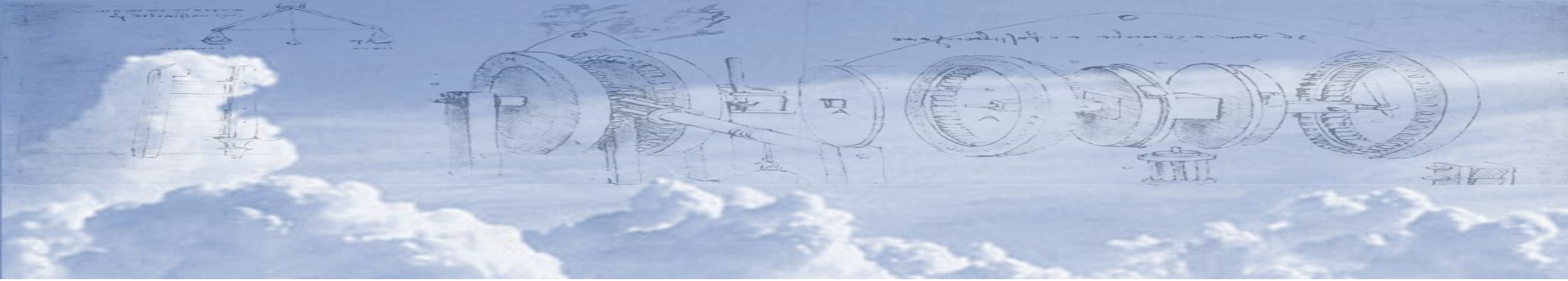
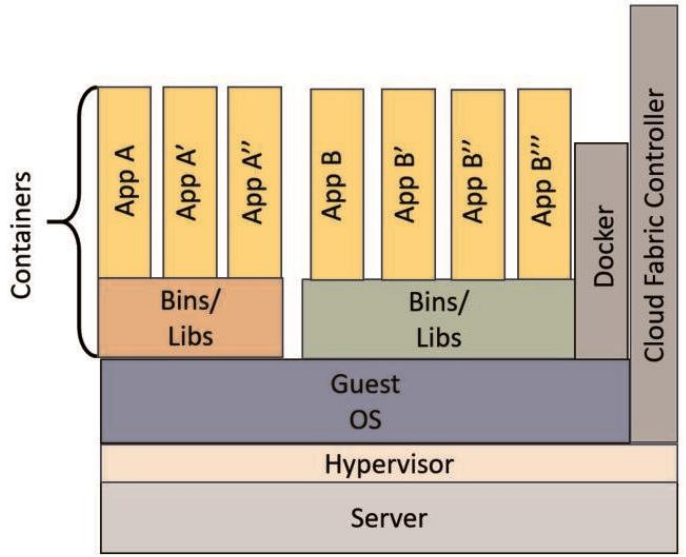


Containers and Microservices

Dennis Gannon

The challenge of software installation and deployment

- Software systems are often complex and installation can be error prone
- When multiple packages need to interact conflicts arise
  - Different software libraries
  - Different operating system versions
- Containerization allows systems to be packaged with everything they need so they can run anywhere.
- Many containers can run on a single system



Docker

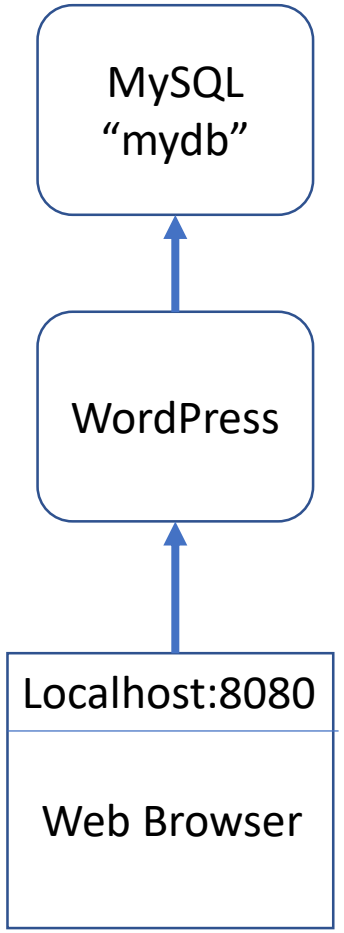
- Several container systems exist, but “Docker” is the most commonly used.
  - It is trivial to install docker on a Mac, PC or Linux
  - Docker Hub is where many applications have been packaged and stored.
  - Running containers can talk to each other.
  - Ran the following on a PC

- Example: to deploy and run a mysql database and then deploy and run an instance of the blog server Wordpress.
- Install MySQL container

$docker run --name mydb –e MYSQL_ROOT_PASSWORD=xxxxx –d mysql

- Now run wordpress container using that instance of MySQL

$docker run --link mydb:mysql –p 8080:80 –d wordpress



MYNEWBLOG

Just another WordPress site

AUGUST 13, 2017 BY DENNIS

myfirst page

here is a blog!

UNCATEGORIZED

Edit

Search ...

RECENT POSTS

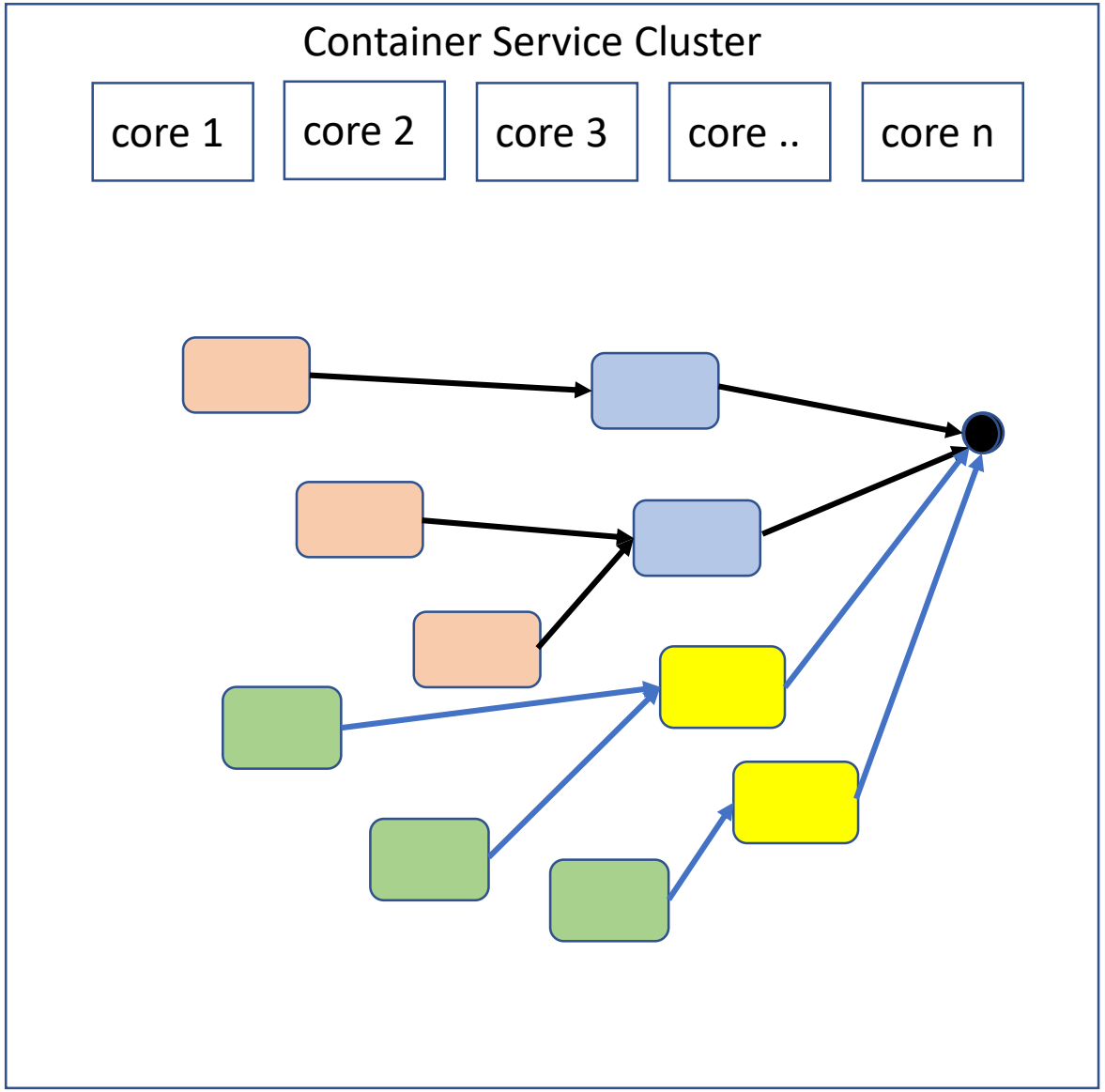
Hello world!

RECENT COMMENTS

A WordPress Commenter on Hello world!

Microservices

- Big on-line services use massive collections of containers to scale
  - Netflix, Google Docs, Azure services, eBay, Amazon, the UK Government Digital Service, Twitter, PayPal, Gilt, Bluemix, Soundcloud, The Guardian
- Divide a computation into small, mostly stateless components that can be
  - Easily replicated for scale
  - Communicate with simple protocols
  - Computation is as a swarm of communicating workers.
- Typically run as containers using a service deployment and management service
  - Amazon EC2 Container Service
  - Google Kubernetes
  - DCOS from Berkeley/Mesosphere
  - Docker Swarm



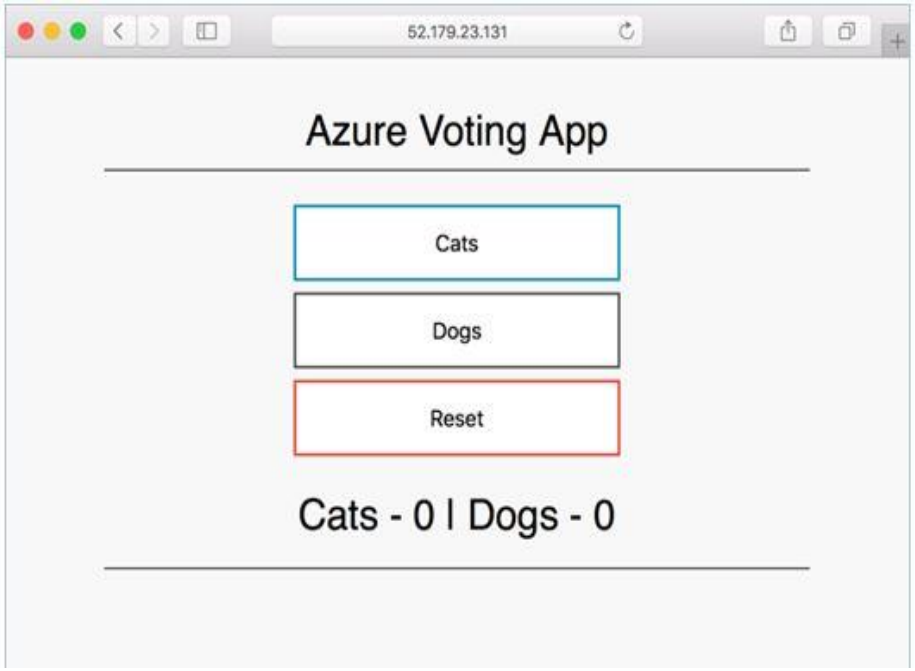
Kubernetes (from Google and now open source)

- Available on Google cloud, AWS and Azure.
- Based on the idea of “pods” of containers where containers in the same pod can easily share resources.
- Creating a cluster on Azure is easy from your laptop with azure cli and the Azure Container Service (acs).

```
>azure login

>az group create --name mygroup  --location eastus

>az acs create --orchestrator-type kubernetes --resource-group mygroup
    --name mykubcluster –generate-ssh-keys

>az acs get-credentials --resource-group=mygroup –name=mykubcluster

>az acs kubernetes browse –g mygroup –n mykubcluster
```

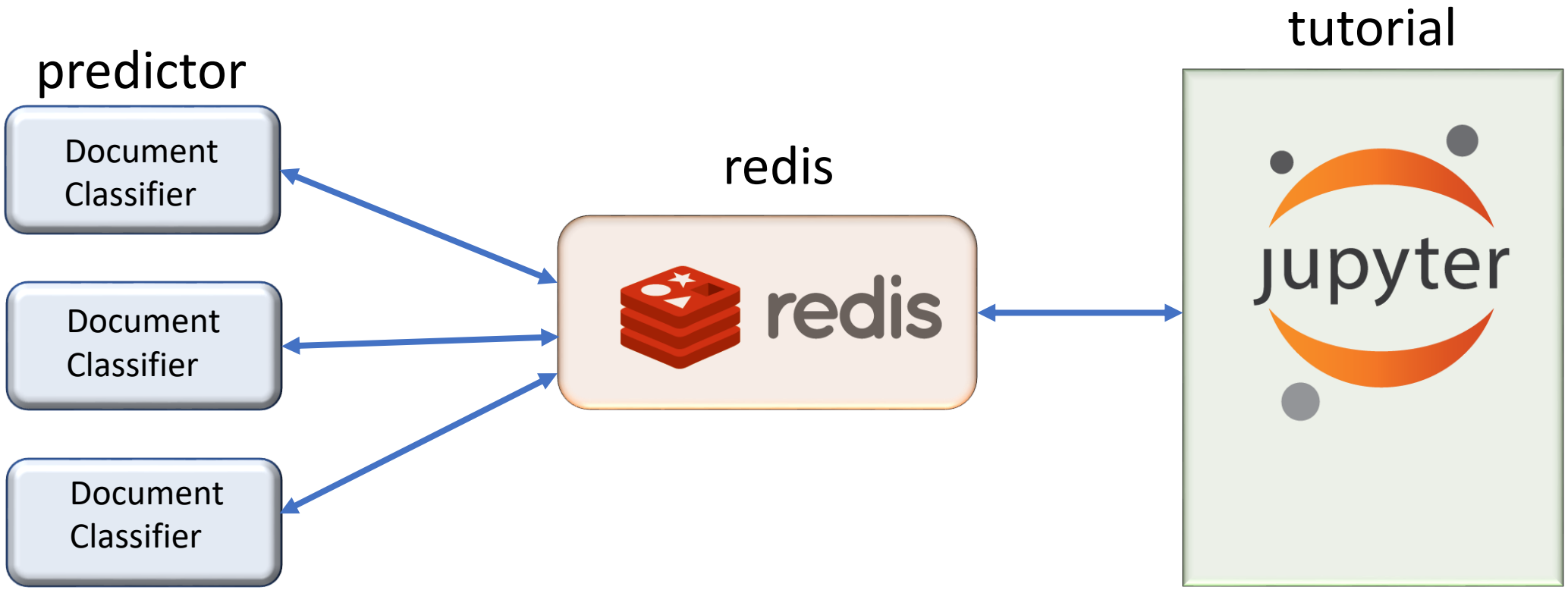
Try Kubernetes yourself

- If we have time do this exercise:
- http://aka.ms/kddacs
  - **Deploy Kubernetes cluster for Linux containers**
- In it you will install Azure command line tools
  - And launch a kubernetes cluster
  - And launch a two container web app.

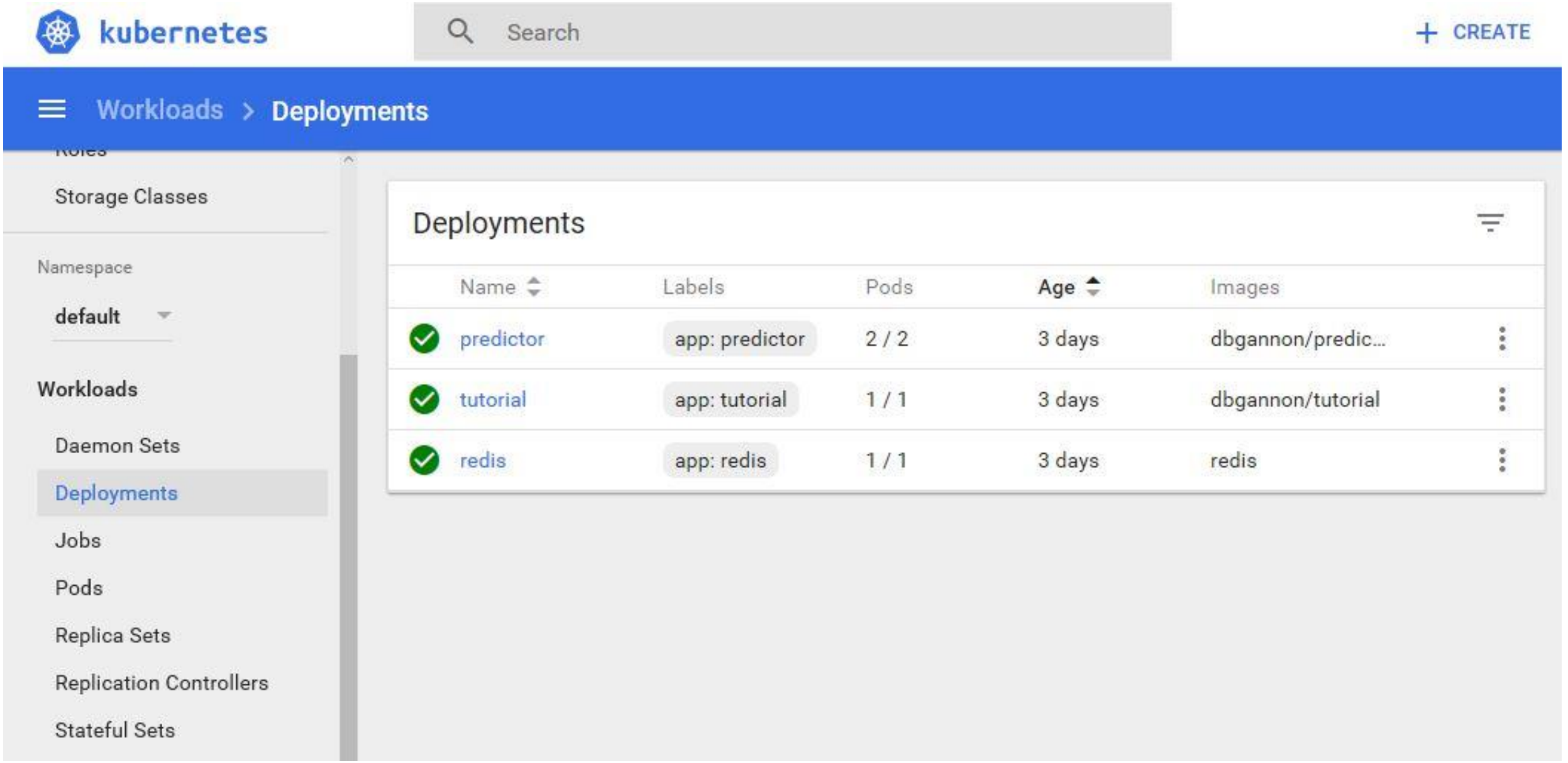


An Example Application

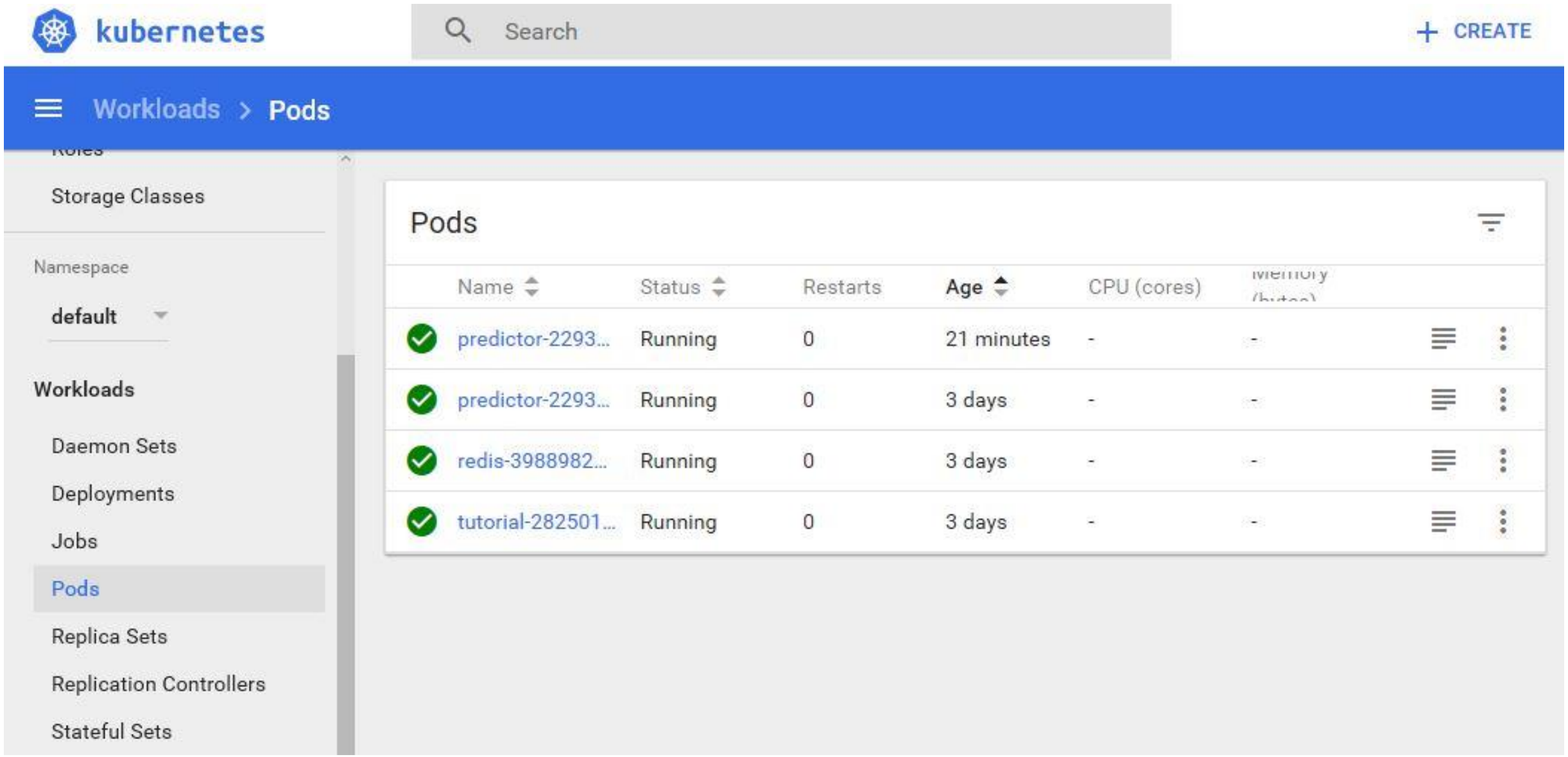
A Predictor container will accept an abstract from a science paper and classify it as either “physics”, “math”, “bio”, “cs” or “finance”. The predictor picks up the documents From an instance of a “redis” cache.
A Jupyter notebook can put documents into the cache and fetch the reply from the classifier.



The Kubernetes Dashboard



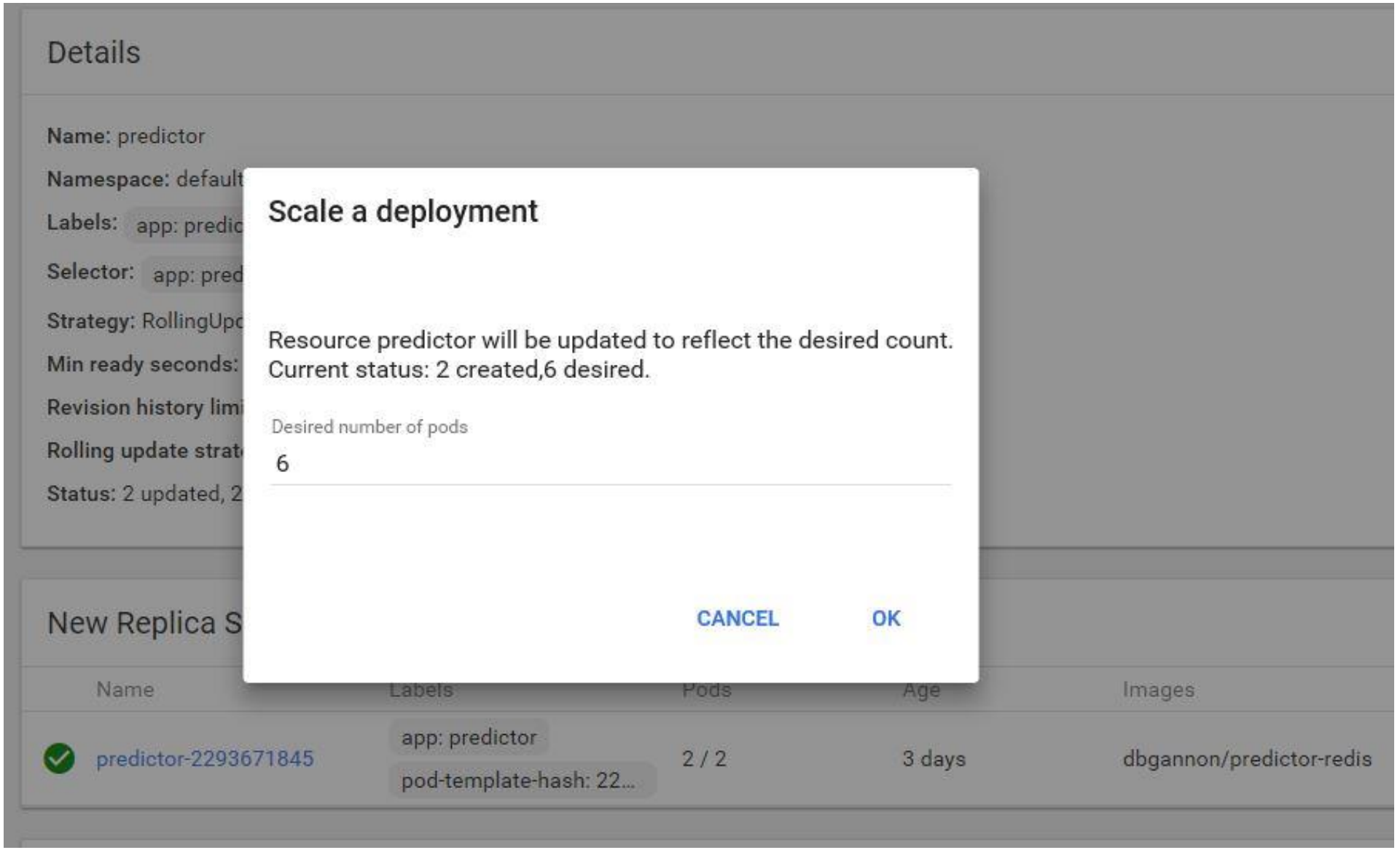
The Currently Running Pods



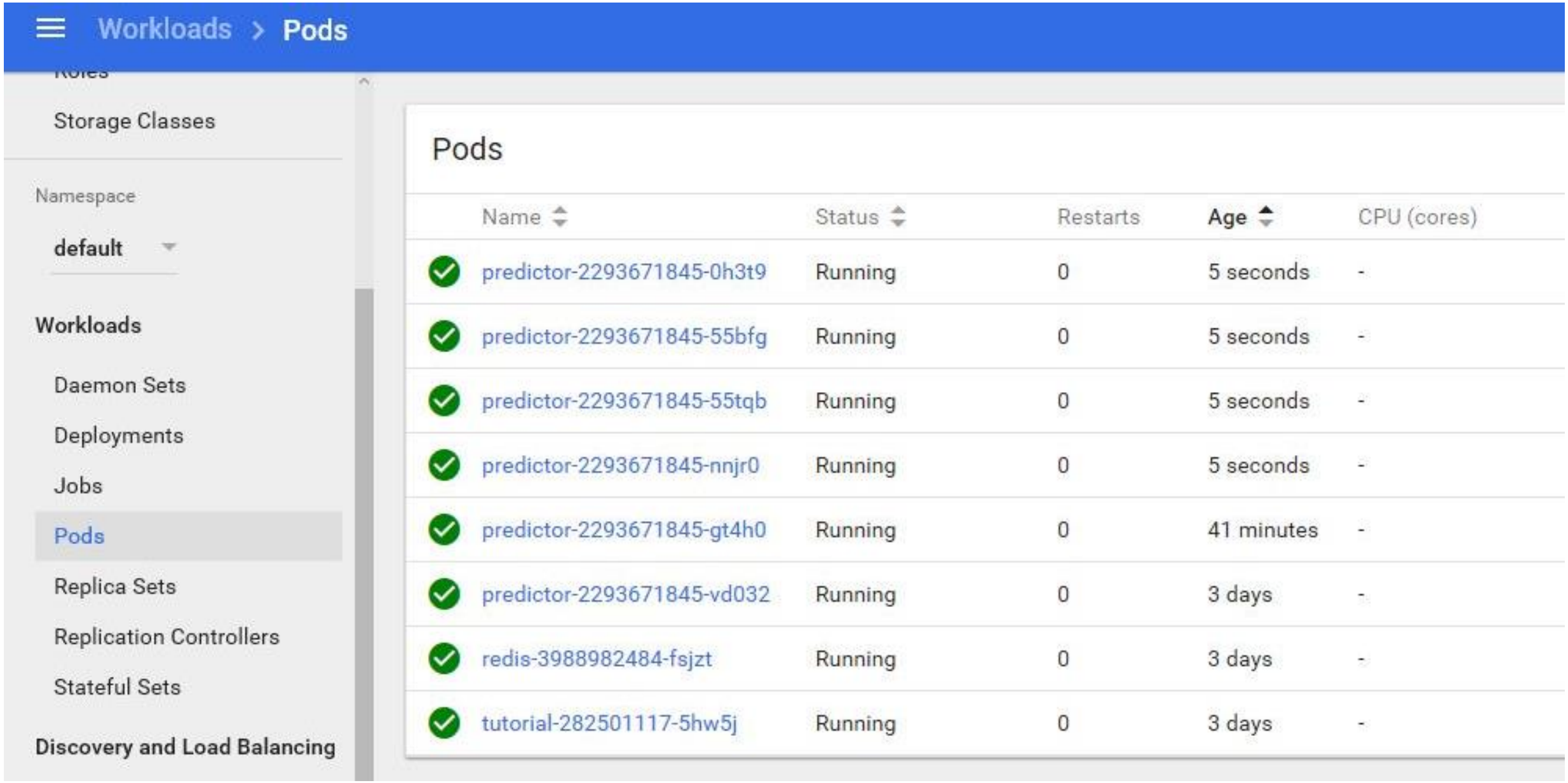
The Python notebook in Jupyter “tutorial”

```
In [68]: print titles[1500]

         New constraints on primordial gravitational waves from Planck 2015 [astro-ph.CO]

In [69]: y = predict.delay(abstracts[1500], titles[1500], sites[1500])

In [71]: y.get()

Out[71]: [[u'Physics', u'Physics', u'Physics', u'Physics', u'Physics']]

In [74]: print titles[164]

         On weighted measure of inaccuracy for doubly truncated random variables [math.ST]

In [77]: y = predict.delay(abstracts[164], titles[164], sites[164])

In [78]: y.get()

Out[78]: [[u'math', u'math', u'math', u'math', u'math']]
```

Let’s increase the number of predictors to 6



Now let’s look at the pods



Conclusion

- Docker containers allow us to deploy complex applications anywhere and without modification.
  - Containers isolate system and library dependencies so that each container is fully portable.
- Kubernetes, Swarm and other orchestrators running on the Azure Container Service make it possible to
  - Create applications with massive numbers of containers.
  - The cluster size for the system can scale to large number of cloud servers
- This is how many of the large services such as Cosmos cloud DB, the Google services, Netflix, etc are built.