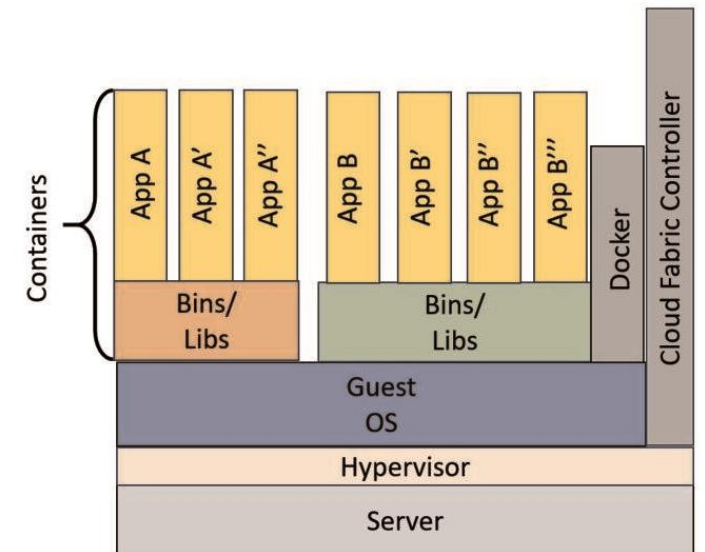


Containers and Microservices

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



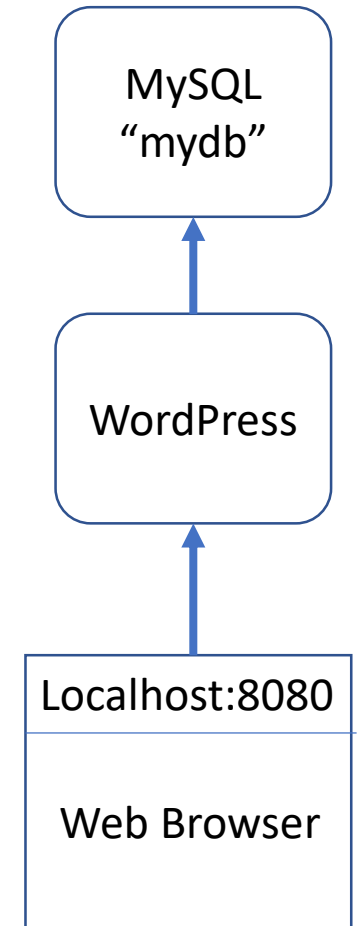
Docker

- 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=xxxxxx -d mysql
```

- Now run wordpress container using that instance of MySQL

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





mynewblog



Customize



1



0



New



Edit Post

Howdy, dennis



MYNEWBLOG

Just another WordPress site

AUGUST 13, 2017 BY DENNIS

myfirst page

here is a blog!



UNCATEGORIZED

Edit

Search ...



RECENT POSTS

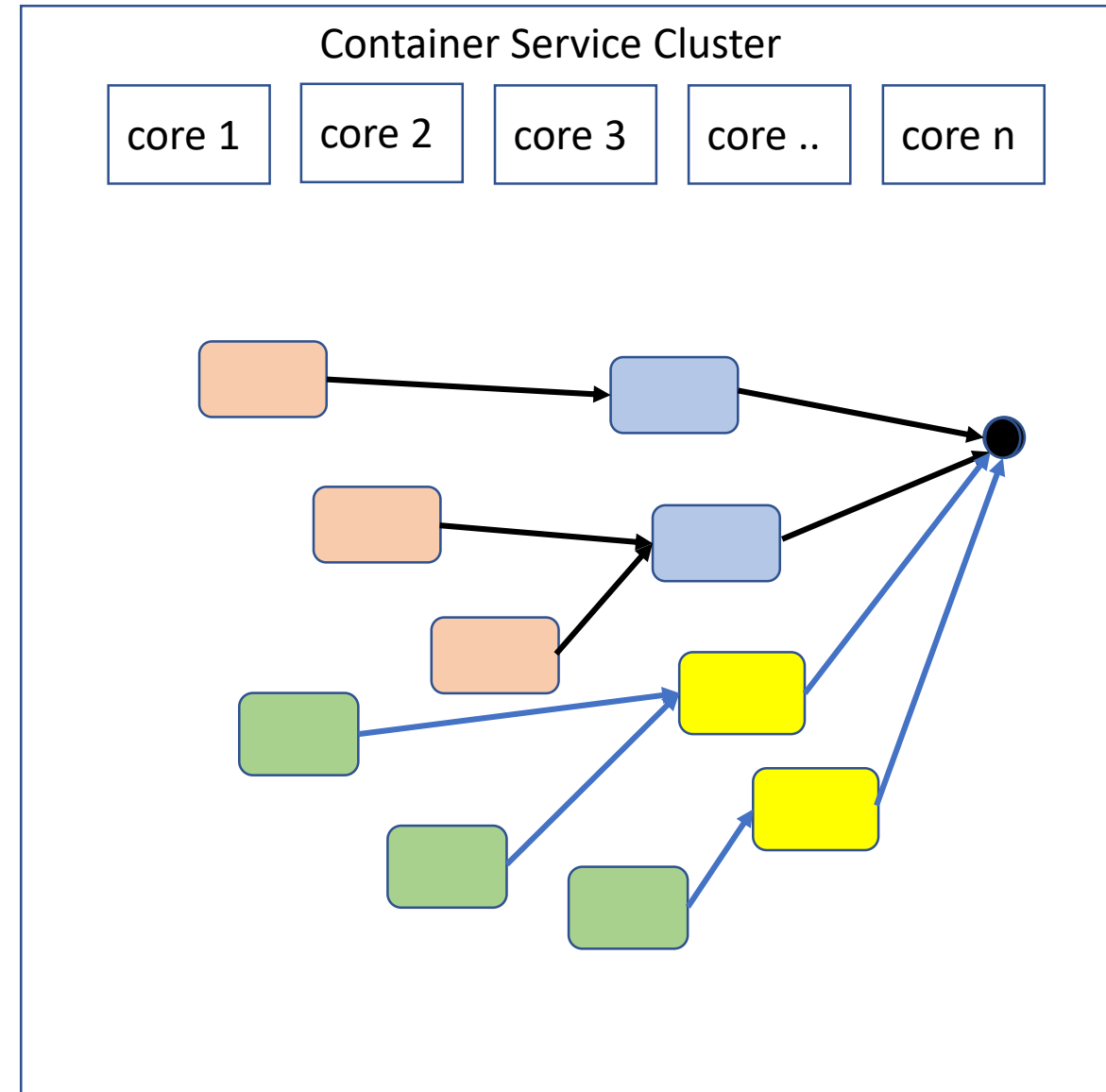
Hello world!

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
```

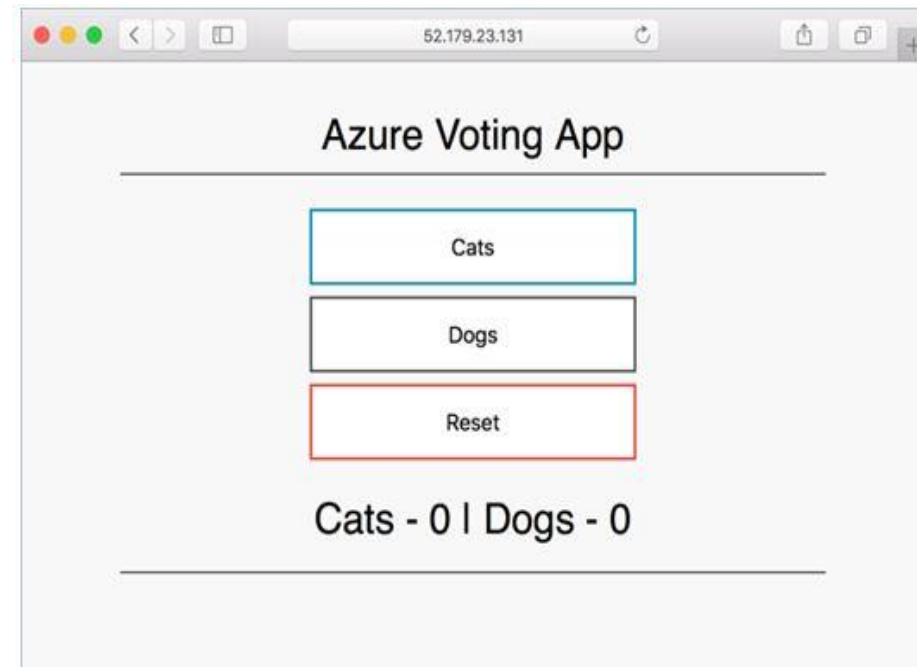
```
>az acs create --orchestrator-type kubernetes --resource-group mygroup  
--name mykubcluster --generate-ssh-keys
```

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

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

Try Kubernetes yourself

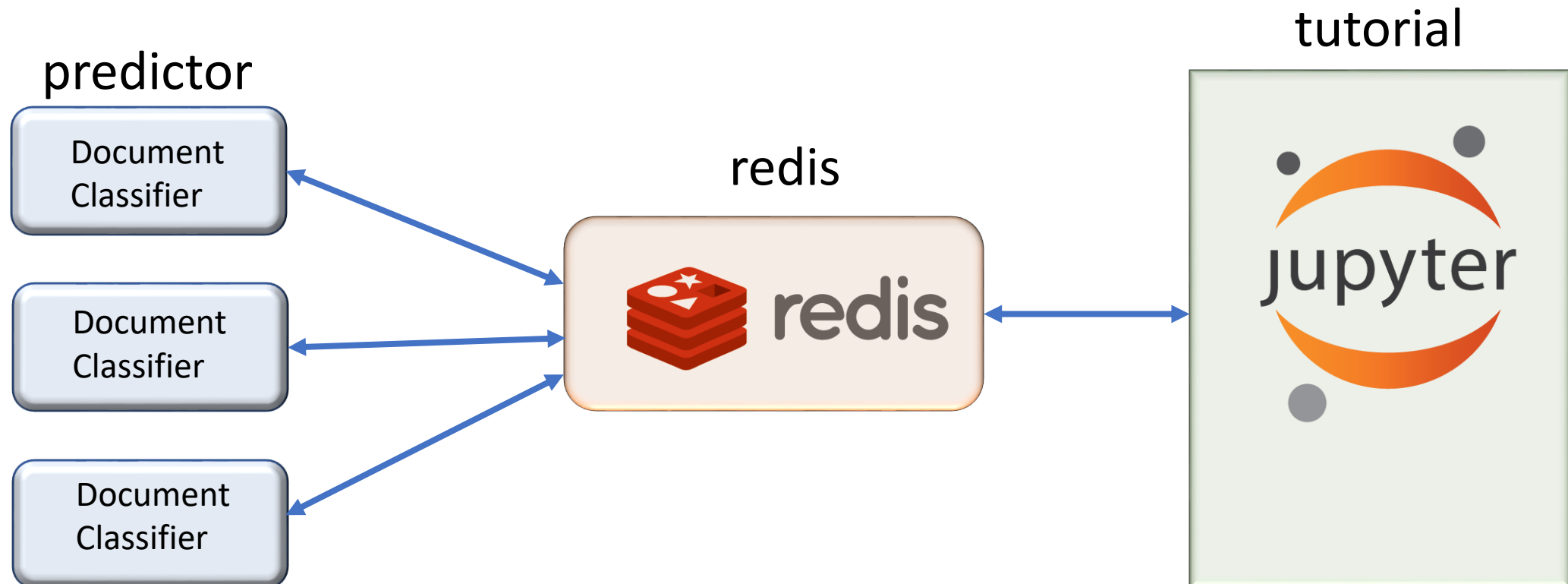
- If we have time do this exercise:
- <http://aka.ms/kddacs>
 - **Deploy Kubernetes cluster for Linux containers**
- In it you will install Azure command line tools
 - And launch a kubernetes cluster
 - And launch a two container web app.



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



Search


+ CREATE

Workloads > Deployments

Deployments

Name	Labels	Pods	Age	Images	
✓ predictor	app: predictor	2 / 2	3 days	dbgannon/predic...	⋮
✓ tutorial	app: tutorial	1 / 1	3 days	dbgannon/tutorial	⋮
✓ redis	app: redis	1 / 1	3 days	redis	⋮

The Currently Running Pods

 **kubernetes**

Search

+ CREATE

Workloads > Pods

Namespaces

Storage Classes

Namespaces

default

Workloads

Daemon Sets

Deployments

Jobs

Pods

Replica Sets

Replication Controllers

Stateful Sets

Pods

	Name	Status	Restarts	Age	CPU (cores)	memory (bytes)		
✓	predictor-2293...	Running	0	21 minutes	-	-	≡	⋮
✓	predictor-2293...	Running	0	3 days	-	-	≡	⋮
✓	redis-3988982...	Running	0	3 days	-	-	≡	⋮
✓	tutorial-282501...	Running	0	3 days	-	-	≡	⋮

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

Details

Name: predictor

Namespace: default

Labels: app: predictor

Selector: app: predictor

Strategy: RollingUpdate

Min ready seconds: 0

Revision history limit: 10

Rolling update strategy: RollingUpdate

Status: 2 updated, 2 created

New Replica Set

Name	Labels	Pods	Age	Images
✓ predictor-2293671845	app: predictor pod-template-hash: 22...	2 / 2	3 days	dbgannon/predictor-redis

Scale a deployment

Resource predictor will be updated to reflect the desired count.
Current status: 2 created, 6 desired.

Desired number of pods
6

CANCEL

OK

Now let's look at the pods

The screenshot shows the Kubernetes dashboard interface. The top navigation bar is blue with a hamburger menu icon and the text "Workloads > Pods". The left sidebar is light gray and contains several sections: "Namespaces" with a dropdown menu showing "default"; "Workloads" with links for "Daemon Sets", "Deployments", "Jobs", "Pods" (which is highlighted), "Replica Sets", "Replication Controllers", and "Stateful Sets"; and "Discovery and Load Balancing". The main content area is white and titled "Pods". It contains a table with the following columns: "Name", "Status", "Restarts", "Age", and "CPU (cores)". There are eight rows of pods, all with a green checkmark icon in the "Name" column and a "Running" status. The pods are: predictor-2293671845-0h3t9, predictor-2293671845-55bfg, predictor-2293671845-55tqb, predictor-2293671845-nnjr0, predictor-2293671845-gt4h0, predictor-2293671845-vd032, redis-3988982484-fsjzt, and tutorial-282501117-5hw5j. The "Restarts" column for all pods shows "0". The "Age" column shows "5 seconds" for the first four pods, "41 minutes" for the fifth, and "3 days" for the last three. The "CPU (cores)" column shows "-" for all pods.

Name	Status	Restarts	Age	CPU (cores)
✓ predictor-2293671845-0h3t9	Running	0	5 seconds	-
✓ predictor-2293671845-55bfg	Running	0	5 seconds	-
✓ predictor-2293671845-55tqb	Running	0	5 seconds	-
✓ predictor-2293671845-nnjr0	Running	0	5 seconds	-
✓ predictor-2293671845-gt4h0	Running	0	41 minutes	-
✓ predictor-2293671845-vd032	Running	0	3 days	-
✓ redis-3988982484-fsjzt	Running	0	3 days	-
✓ tutorial-282501117-5hw5j	Running	0	3 days	-

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.