



Deploying a custom Hyrax on Azure with Terraform and Kubernetes from a standing start

-Julie ~~Rory~~ McNicholl

Docker Images deployed to a ...
Kubernetes Cluster hosted on ...
Azure Cloud Services built using
Terraform Infrastructure Creation

Azure Resource Group



Azure Automation Account

Azure Log Analytics Workspace

Azure Kubernetes Cluster

fcrepo

postgresdb

solr

app_web

app_worker

redis

Azure Storage Account

Azure Storage: Files

Azure Storage: Files

Azure Managed Disk

/data

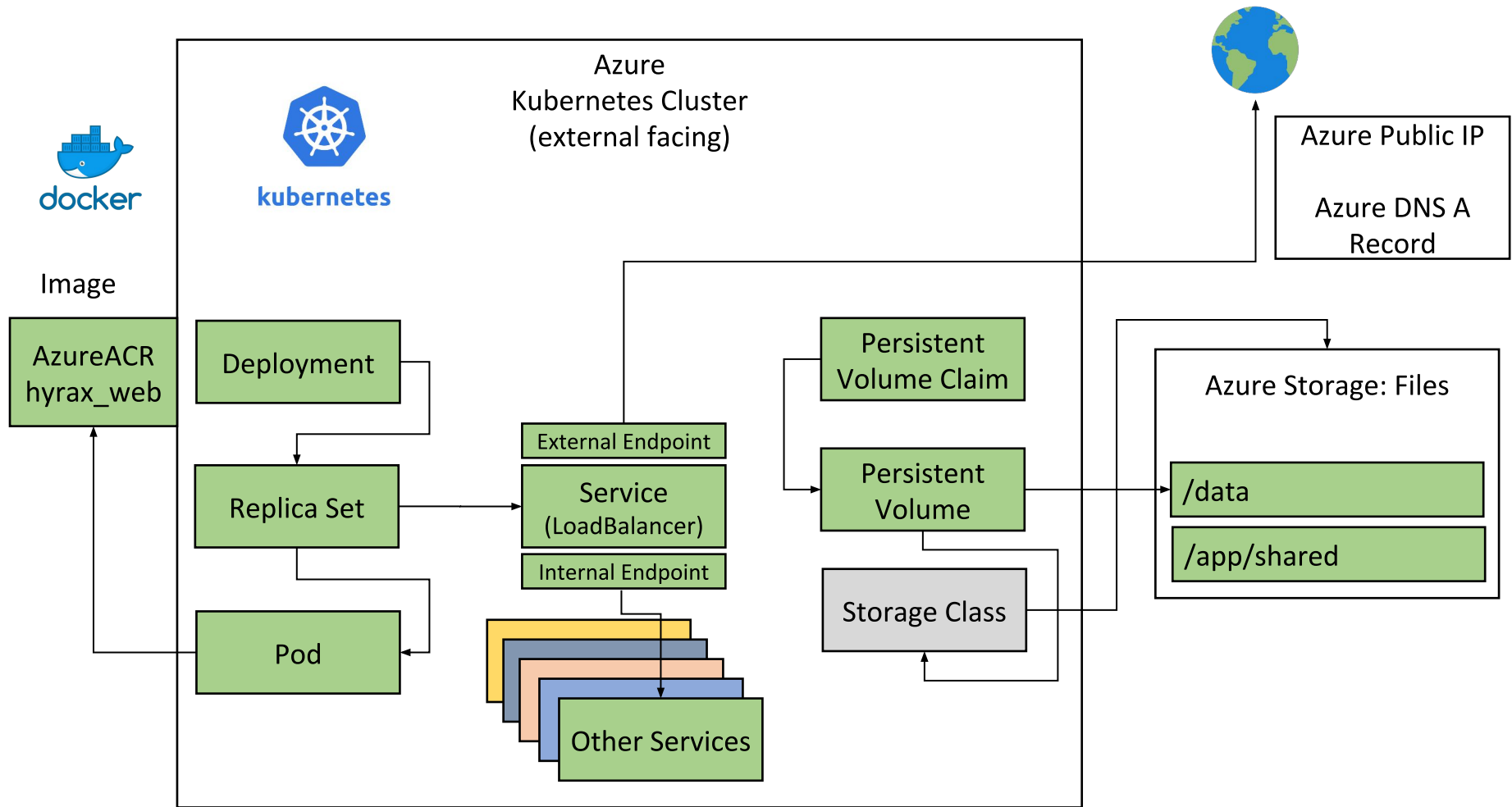
/var/lib/postgresql/data

/opt/solr/server/solr/mycores
/data

/data
/app/shared/state

/data







Create Resource Group 

Create Storage Account 

Create Storage Share 

Create Container Registry (ACR) 

Tag and Push Docker Images to ACR  → 

Create Automation Account 

Create Log Analytics Workspace 

Create Cluster 

Create Secrets  → 

Create Public IPs 

Create DNS Zone A Record 

Create Storage Classes, PVCs and PVs  → 

Copy Solr Config to solr:/data  → 

Create Deployments and Services 

Create Postgresdb, Solr, Redis  ← 

Create Fcrepo  ← 

Create Web and Worker  ← 

Manual (pre terraform on local CLI)

Build Docker Image for the Hyrax App using
`docker-compose build`

Manual (post terraform on Azure Portal)

Deploy VM Start|Stop Solution in Automation Account

@todo

- Use remote-state
- Make running docker-compose build part of the terraform apply
 - Delay kubernetes deploy for hyrax/sidekiq until it's finished
 - Initiate only on a change in git revision/tag
 - Circle CI?
- Endpoint Security
 - SSL for the public endpoint(s) ✓