



# Administering Banner Applications in Docker Containers

Gabriel Tocci

09-OCT-2019

11:15am - 12:15pm



---

EAST TENNESSEE STATE  
UNIVERSITY

---

# Session Format

- 60 minute time slot
  - Touch on a lot of topics
    - WHY Docker (brief)
    - HOW Docker (demo)
  - Q&A anytime
- Can you see REAL good?
  - Showing Code
  - Demo
- [gabrieltozzi.com/talks](http://gabrieltozzi.com/talks)



# Configuration Management & IaC Deployment Automation (CD)


- Configuration management
  - Ansible, Puppet, Chef, etc.
- Image-based management
  - VM Cloning, Packer, Vagrant, etc.
- Containerization
  - Docker, rkt, mesos

## Benefits

- Manages “drift” of configurations
- Declare intent and interactions of resources
- Auditable infrastructure
- Increases recovery speed
- Reduces go-live errors

Whatever the approach, configurations should be:

- Documented
- Repeatable
- Codified
- Automated

Code =  git

## ETSU Toolset

- Puppet – R10k
- Terraform
- Docker
- Gitlab-CI



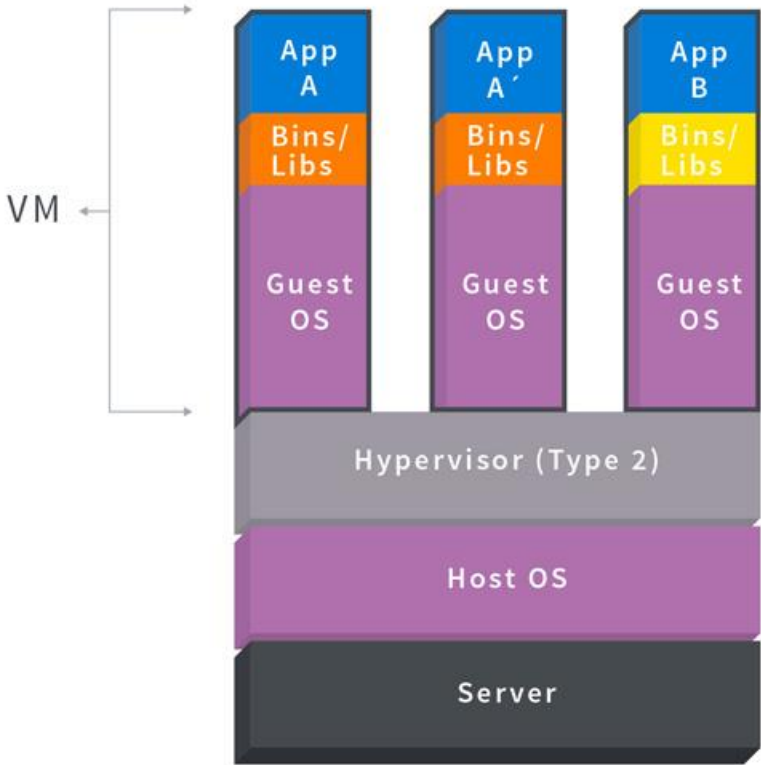
# Containers: Applications **contain** their dependencies

Deployments are defined by the management of their dependencies:

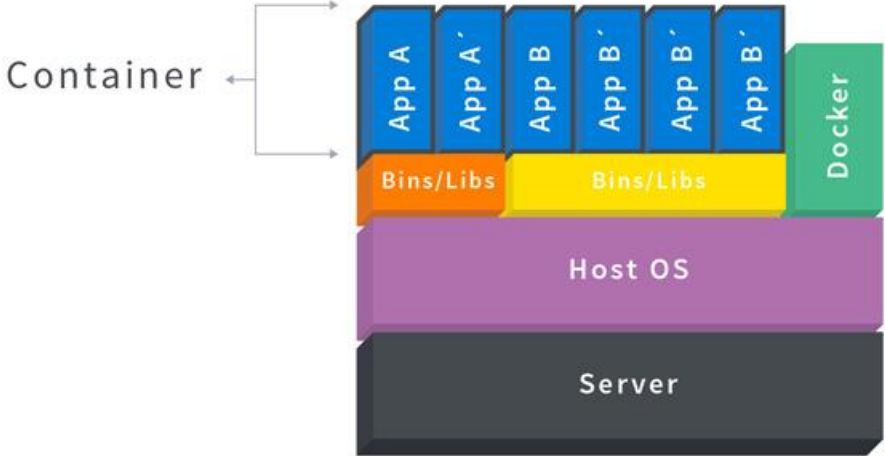
- Operating System Flavor (CentOS, RHEL, OEL, etc.)
- OS Packages (libaio, openssl, curl, etc.)
- OS Configuration (limits, accounts, iptables, etc.)
- Application Server & Version (Tomcat 7/8)
- Application Service Configuration (data sources / connection strings, secrets, etc.)



# Containers vs. VMs



Containers are isolated, but share OS and, where appropriate, bins/libraries



# Base Images v. Baked Images

- Existing images can be used as starting points for other images:  
**base images**
- Images share **cached layers**
- Configuring runtime parameters at container startup avoids “**baking**” configurations into the image
- Arguments can be passed to containers via the run command or (more commonly) through ENV variables

To override the configuration that was added into the WAR file, you must set system properties to point to external configuration files. For example, to point to a configuration file residing in the `PRODUCT_HOME` directory, export `JAVA_OPTS=`  
`"-DBANNER_APP_CONFIG=/PRODUCT_HOME/shared_configuration/  
banner_configuration.groovy  
-DBANNER_STUDENT_REGISTRATION_SSB_CONFIG=/PRODUCT_HOME/  
StudentRegistrationSsb/current/instance/config/  
StudentRegistrationSsb_configuration.groovy"`.



# Base Tomcat Image

```
Dockerfile 505 Bytes

1 FROM tomcat:8-jre7
2 MAINTAINER Ian Dillon <dillon@etsu.edu>
3
4 # Make sure AWS cli is installed.
5 RUN apt-get update && apt-get install -y \
6     python-pip \
7     && pip install awscli \
8     && rm -rf /var/lib/apt/lists/* \
9     && apt-get clean \
10    && rm -fr /usr/local/tomcat/webapps/*
11
12 # Move oci8 / db libs into place
13 ADD ojdbc6.jar /usr/local/tomcat/lib/
14 ADD xdb6.jar /usr/local/tomcat/lib/
15
16 COPY entrypoint.py /usr/local/tomcat/bin/
17 ENTRYPOINT ["/usr/bin/python", "/usr/local/tomcat/bin/entrypoint.py"]
```



# Gitlab-ci.yml



Amazon ECR

```
1  stages:
2  |  - deploy
3
4  deploy_dev:
5  |  - stage: deploy
6  |  - script:
7  |  |  - "`aws ecr get-login --region $AWS_REGION`"
8  |  |  - docker build -t $AWS_ECR_URL/$CI_PROJECT_NAME:dev .
9  |  |  - docker push $AWS_ECR_URL/$CI_PROJECT_NAME:dev
10 |  - only:
11 |  |  - branches
12 |  - except:
13 |  |  - master
14
15 deploy_prod:
16 |  - stage: deploy
17 |  - script:
18 |  |  - "`aws ecr get-login --region $AWS_REGION`"
19 |  |  - docker build -t $AWS_ECR_URL/$CI_PROJECT_NAME:prod .
20 |  |  - docker push $AWS_ECR_URL/$CI_PROJECT_NAME:prod
21 |  - environment:
22 |  |  - name: production
23 |  - only:
24 |  |  - master
25 |  - when: manual
26
```





# Custom Entrypoint

- Python
  - Easy to use
  - Libraries to do what we need:
    - OS Interaction, File Operations, YAML Config Parsing
- Pull in per-environment configurations
  - Secrets: S3
  - Configuration files
  - Application
- kv replacement of secrets in config files at deployment
- Specify the startup process
  - **Tomcat:** `os.system("/usr/local/tomcat/bin/catalina.sh run")`
  - **CentOS:** `os.system("/usr/local/bin/start.sh")`










# Tomcat entrypoint.py

```
15
16 config_base = '/tmp/config/'
17 secrets_file = 'secrets.yaml'
18
19 if os.getenv("S3_CONFIG_TAR") is not None:
20     os.system("mkdir -p " + config_base + " && aws s3 cp $S3_CONFIG_TAR - | tar -C " + config_base + " -x")
21
22 if os.getenv("S3_SECRETS") is not None:
23     os.system("mkdir -p " + config_base + " && aws s3 cp $S3_SECRETS " + config_base)
24
25 config = ConfigParser.ConfigParser({"has_secrets": "false"})
26 config.read(config_base + "docker.ini")
27 secrets = yaml.load( open(config_base + secrets_file, "r") )
28
29 for section in config.sections():
30     source = config_base + section
31     destination = config.get(section, "destination") + section
32     if config.getboolean(section, "has_secrets") is True:
33         print "Replacing secrets in " + section
34         with open(source) as infile, open(destination, "w") as outfile:
35             for line in infile:
36                 for k, v in secrets.iteritems():
37                     line = line.replace(k, v)
38                 outfile.write(line)
39     else:
40         shutil.copy(source, destination)
41
42 os.system("/usr/local/tomcat/bin/catalina.sh run")
43 exit()
```

# Config Repo



Name	Last commit
..	
 StudentSelfService.war	TBR Q3 2018 StudentSelfService PROD upgrade
 StudentSelfService_configuration.groovy	Adding Banner 9 SelfServ PROD
 banner_configuration.groovy	Adding Banner 9 SelfServ PROD
 context.xml	Adding Banner 9 SelfServ PROD
 docker.ini	Adding Banner 9 SelfServ PROD
 server.xml	Updated database host: etsupdb->pdb.infosys
 setenv.sh	Adding Banner 9 SelfServ PROD



# Config: docker.ini

docker.ini 390 Bytes

```
1 [StudentSelfService.war]
2 destination = /usr/local/tomcat/webapps/
3
4 [context.xml]
5 destination = /usr/local/tomcat/conf/
6
7 [server.xml]
8 destination = /usr/local/tomcat/conf/
9 has_secrets = true
10
11 [setenv.sh]
12 destination = /usr/local/tomcat/bin/
13
14 [banner_configuration.groovy]
15 destination = /usr/local/tomcat/conf/
16
17 [StudentSelfService_configuration.groovy]
18 destination = /usr/local/tomcat/conf/
```

```
36 -->
37 <GlobalNamingResources>
38 <!-- Editable user database that can also be used by
39      UserDatabaseRealm to authenticate users
40 -->
41 <Resource name="UserDatabase" auth="Container"
42           type="org.apache.catalina.UserDatabase"
43           description="User database that can be updated and saved"
44           factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
45           pathname="conf/tomcat-users.xml" />
46 <Resource name="jdbc/bannerDataSource"
47           auth="Container" type="javax.sql.DataSource"
48           driverClassName="oracle.jdbc.OracleDriver"
49           url="jdbc:oracle:thin:@//pdb.infosys.etsu.edu:1521/PROD.ETSU.EDU"
50           username="banproxy"
51           password="BAN_TIER_6"
52           initialSize="5"
53           maxTotal="100"
54           maxIdle="-1"
55           maxWaitMillis="30000"
56           validationQuery="select 1 from dual"
57                       accessToUnderlyingConnectionAllowed="true"
58           testOnBorrow="true"
59           testWhileIdle="true"
60           timeBetweenEvictionRunsMillis="300000">
61 </Resource>
```



# Gitlab-ci.yml



Amazon S3

```
1  stages:
2  |  - - - deploy
3
4  deploy:
5  |  - - - stage: deploy
6  |  - - - script:
7  |  |  - - - pip install boto3
8  |  |  - - - python deploy_to_s3.py
9  |  - - - artifacts:
10 |  |  - - - untracked: true
11 |  |  - - - expire_in: 1 mos
12
```



```

1 import os
2 import tarfile
3 import boto3
4 import botocore
5 import filecmp
6
7 BUCKET = 'etsuis-ecs-config'
8 SERVICE = 'BannerGeneralSsb'
9 instances = ['tsth', 'bant', 'prod']
10
11 s3 = boto3.resource('s3')
12
13 for inst in instances:
14     s3_filename = SERVICE+'.tar'
15     tar_filename = SERVICE+'-'+inst+'.tar'
16     curr_filename = SERVICE+'-'+inst+'-current.tar'
17
18     #Create new tar:
19     with tarfile.open(tar_filename, mode='w') as archive:
20         archive.add('./'+inst+'/', arcname='.')
21
22     #Get current S3 tarball:
23     key = inst+'/'+s3_filename
24     try:
25         s3.Bucket(BUCKET).download_file(key, curr_filename)
26     except botocore.exceptions.ClientError as e:
27         if e.response['Error']['Code'] == "404":
28             open(curr_filename, 'a').close() #Just create an empty file
29
30     if not filecmp.cmp(tar_filename, curr_filename):
31         print 'Tarballs for '+ inst + ' differ, uploading new tarball.'
32         s3.Bucket(BUCKET).upload_file(tar_filename, key)
33     else:
34         # Files are the same, so rm them so they don't get archived as CI artifacts
35         os.remove(tar_filename)
36         os.remove(curr_filename)
37

```





Amazon ECR

## Base Image

- OS
- OS Dependencies
- Middleware
- Libs
- Entrypoint Script



Amazon S3

## Tarball

- Application
- Configuration Files

## Secrets File



# Kubernetes





# Container Orchestration

- Determine what containers run:
  - When? Where? How Many?
- What is the deployment strategy: *spread* v. *binpack*
- Enforce resource limits (CPU, RAM) on running containers and for deploying new containers
- Cloud based or on-premise
- Manage or a harness for metrics, logging, etc.
- May manage mesh network, shared volumes, etc.



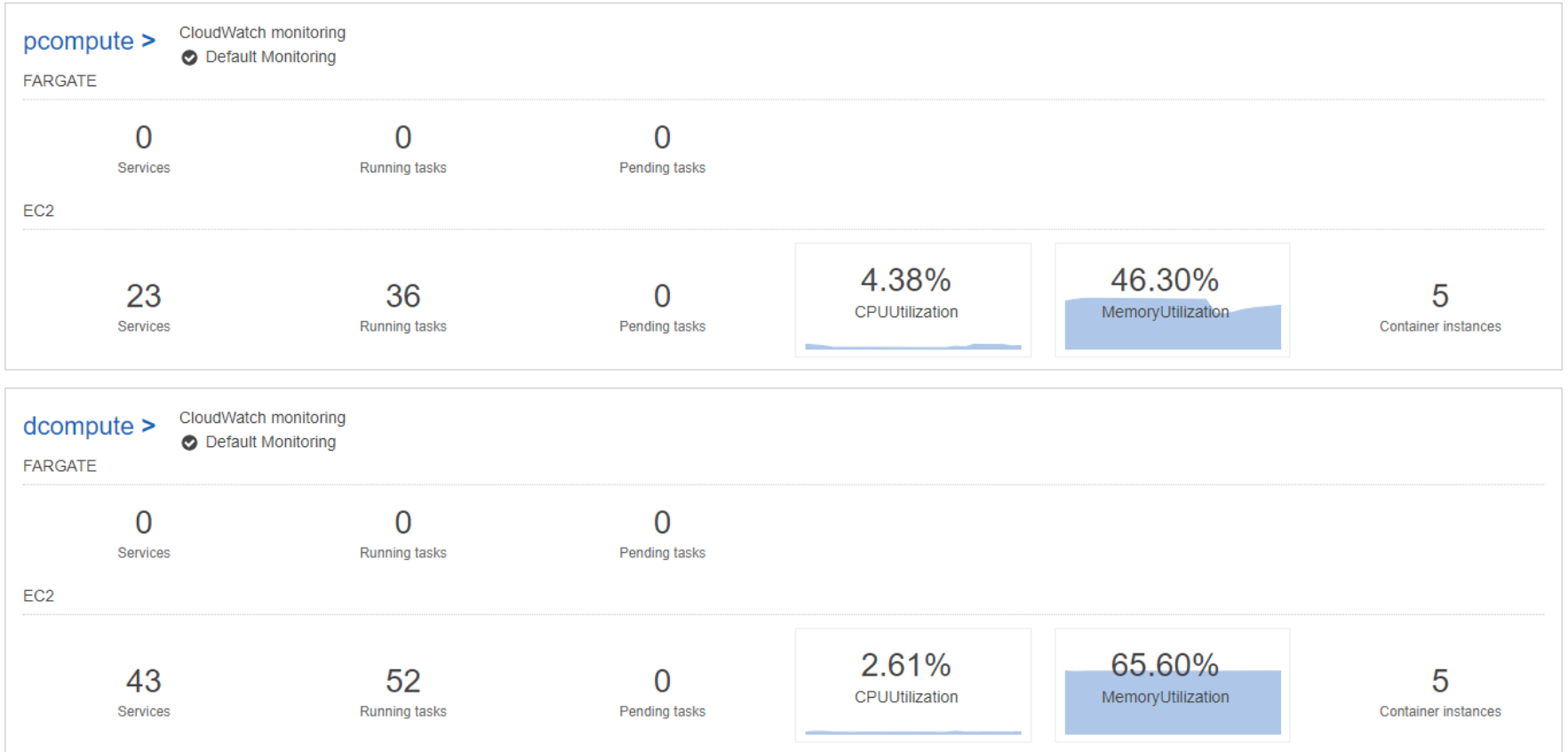
# AWS Elastic Container Service



- Managed Cloud Service (Paas)
- Docker Clusters
  - High Availability – Multi AZ
  - Scalable
  - VMs (EC2) v. Fargate
- Scheduled Tasks
- AWS Integrations
  - Elastic Container Registry (ECR)
  - Access Management (IAM)
  - Logging & Alerting (Cloudwatch)
  - Load Balancer (ALB)



# 2 ECS Clusters PROD (pcompute) & DEV (dcompute)



Registered container instances 3

Pending tasks count 0 Fargate, 0 EC2

Running tasks count 0 Fargate, 28 EC2

Active service count 0 Fargate, 20 EC2

Draining service count 0 Fargate, 0 EC2

- Services
- Tasks
- ECS Instances
- Metrics
- Scheduled Tasks
- Tags

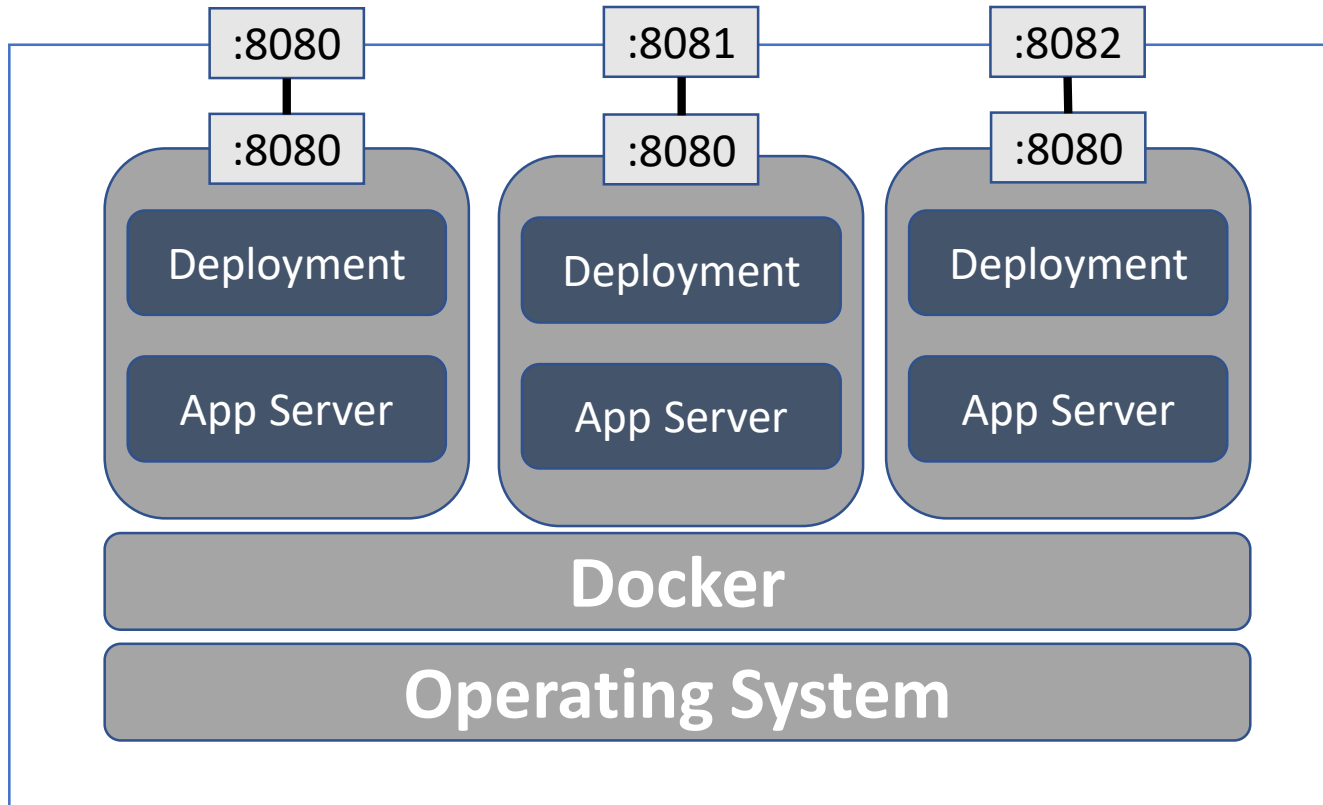
- Create
- Update
- Delete
- Actions

Last updated on March 27, 2019 12:53:31 PM (1m ago)

Filter in this page    Launch type ALL    Service type ALL

<input type="checkbox"/>	Service Name	Status	Service type	Task Definition	Desired tasks	Running tasks	Launch type	Platform version
<input type="checkbox"/>	pcompute-nodeExporter	ACTIVE	REPLICA	pcompute-nodeExport...	3	3	EC2	--
<input type="checkbox"/>	prod-BannerExtensibility	ACTIVE	REPLICA	prod-BannerExtensibil...	1	1	EC2	--
<input type="checkbox"/>	prod-StudentSelfService	ACTIVE	REPLICA	prod-StudentSelfServi...	1	1	EC2	--
<input type="checkbox"/>	prod-dgwAdmin	ACTIVE	REPLICA	prod-dgwAdmin:1	1	1	EC2	--
<input type="checkbox"/>	prod-d2I-Idi	ACTIVE	REPLICA	prod-d2I-Idi:3	1	1	EC2	--
<input type="checkbox"/>	prod-EmployeeSelfService	ACTIVE	REPLICA	prod-EmployeeSelfSe...	1	1	EC2	--
<input type="checkbox"/>	prod-csched	ACTIVE	REPLICA	prod-csched:1	1	1	EC2	--
<input type="checkbox"/>	prod-BannerAdmin	ACTIVE	REPLICA	prod-BannerAdmin:2	2	2	EC2	--
<input type="checkbox"/>	prod-tcommerce	ACTIVE	REPLICA	prod-tcommerce:1	1	1	EC2	--
<input type="checkbox"/>	prod-odsconsole	ACTIVE	REPLICA	prod-odsconsole:1	1	1	EC2	--
<input type="checkbox"/>	prod-RoboRegistrar	ACTIVE	REPLICA	prod-RoboRegistrar:1	1	1	EC2	--
<input type="checkbox"/>	prod-ssb	ACTIVE	REPLICA	prod-ssb:2	2	2	EC2	--
<input type="checkbox"/>	prod-dgwAPIs	ACTIVE	REPLICA	prod-dgwAPIs:1	1	1	EC2	--
<input type="checkbox"/>	prod-goldlink	ACTIVE	REPLICA	prod-goldlink:2	2	2	EC2	--
<input type="checkbox"/>	pcompute-cAdvisor	ACTIVE	REPLICA	pcompute-cAdvisor:1	3	3	EC2	--
<input type="checkbox"/>	prod-dgwTess	ACTIVE	REPLICA	prod-dgwTess:1	1	1	EC2	--
<input type="checkbox"/>	prod-appNav	ACTIVE	REPLICA	prod-appNav:1	2	2	EC2	--
<input type="checkbox"/>	prod-dgwComposer	ACTIVE	REPLICA	prod-dgwComposer:1	1	1	EC2	--
<input type="checkbox"/>	prod-dgwDashboard	ACTIVE	REPLICA	prod-dgwDashboard:1	1	1	EC2	--
<input type="checkbox"/>	prod-drup	ACTIVE	REPLICA	prod-drup:1	1	1	EC2	--

# Container Networking



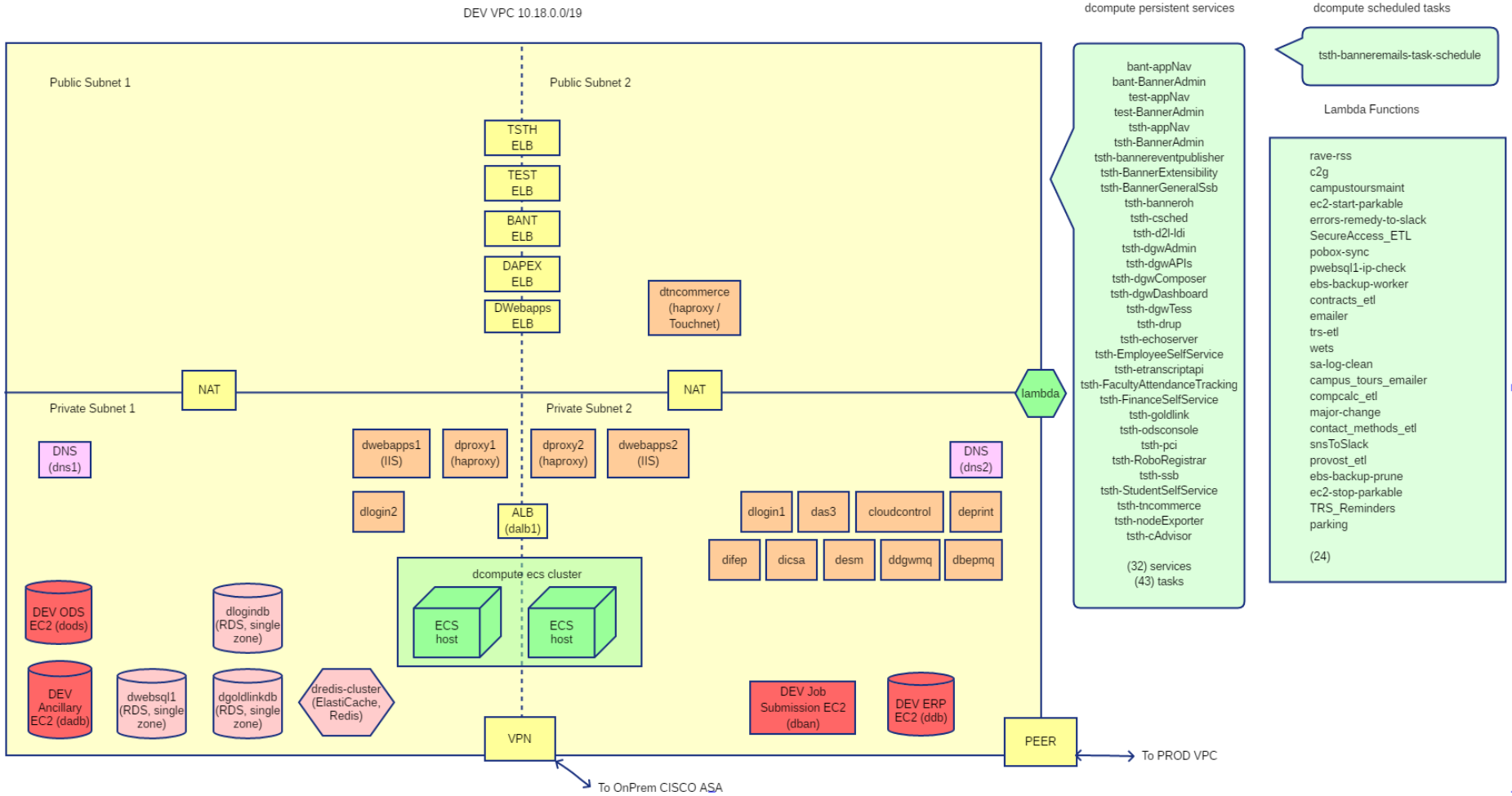


To edit, select a mode above.

dalb1 | HTTP:80 (52 rules)

▶ Rule limits for condition values, wildcards, and total rules.

1	arn...c5e07 ▼	<b>IF</b> ✓ Path is /BannerExtensibility*	<b>THEN</b> Forward to <a href="#">tsth-BannerExtensibility</a>
2	arn...7b590 ▼	<b>IF</b> ✓ Host is tsth.infosys.etsu.edu ✓ Path is /bannerHelp*	<b>THEN</b> Forward to <a href="#">tsth-BannerAdmin</a>
3	arn...34dc7 ▼	<b>IF</b> ✓ Host is tsth.infosys.etsu.edu ✓ Path is /BannerAdmin.ws*	<b>THEN</b> Forward to <a href="#">tsth-BannerAdmin</a>
4	arn...64d0e ▼	<b>IF</b> ✓ Path is /BannerAdmin* ✓ Host is tsth.infosys.etsu.edu	<b>THEN</b> Forward to <a href="#">tsth-BannerAdmin</a>
5	arn...d2463 ▼	<b>IF</b> ✓ Host is tsth.infosys.etsu.edu ✓ Path is /StudentSelfService*	<b>THEN</b> Forward to <a href="#">tsth-StudentSelfService</a>
6	arn...a8570 ▼	<b>IF</b> ✓ Path is /EmployeeSelfService* ✓ Host is tsth.infosys.etsu.edu	<b>THEN</b> Forward to <a href="#">tsth-EmployeeSelfService</a>
7	arn...48cce ▼	<b>IF</b> ✓ Path is /applicationNavigator* ✓ Host is tsth.infosys.etsu.edu	<b>THEN</b> Forward to <a href="#">tsth-appNav</a>



# Deployment Process

## New Application (install)

- New ECS Task definition: Terraform
  - Base image, memory, networking, ENV (config tar & secrets file), log group
- Add service to ECS: Terraform
  - Cluster, task, count, security role, ALB target, placement strategy
- Add new route to haproxy for path to ALB
- Create ecs repository: Git
  - war, config files, docker.ini

## Existing Application (upgrade)

- Commit new war and config files to ecs repository: git
- Restart ecs task: AWS Console





# ECS Task



ECS

```
1  [
2  {
3    "name": "prod-BannerAdmin",
4    "image": "ACCOUNT.dkr.ecr.us-east-1.amazonaws.com/tomcat8-jre7-base:dev",
5    "memoryReservation": 8192,
6    "essential": true,
7    "networkMode": "bridge",
8    "portMappings": [
9      {
10     "hostPort": 0,
11     "containerPort": 8080,
12     "protocol": "tcp"
13   }
14 ],
15   "environment": [
16     {
17       "name": "S3_CONFIG_TAR",
18       "value": "s3://BUCKET_NAME/prod/AdminPages.tar"
19     },
20     {
21       "name": "S3_SECRETS",
22       "value": "s3://BUCKET_NAME/prod/secrets.yaml"
23     }
24   ],
25   "logConfiguration": {
26     "logDriver": "awslogs",
27     "options": {
28       "awslogs-group": "prod-BannerAdmin",
29       "awslogs-region": "us-east-1",
30       "awslogs-stream-prefix": "us-east-1"
31     }
32   }
33 }
34 ]
```



# ECS Service



```
resource "aws_ecs_service" "prod-BannerAdmin" {  
  name = "prod-BannerAdmin"  
  cluster = "${aws_ecs_cluster.pcompute.id}"  
  task_definition = "${aws_ecs_task_definition.prod-BannerAdmin.arn}"  
  desired_count = 2  
  iam_role = "${aws_iam_role.pecs_service_role.arn}"  
  depends_on = ["aws_iam_role_policy.pecs_service_role_policy"]  
  load_balancer {  
    target_group_arn = "${aws_alb_target_group.prod-BannerAdmin.arn}"  
    container_name = "prod-BannerAdmin",  
    container_port = 8080  
  }  
  .....  
}
```



# ELB/HAProxy

```
prod/pproxy1/haproxy.cfg
@@ -208,6 +208,12 @@ frontend main
208 208     use_backend prod-EmployeeSSB if src-etsu-wifi host-banner-infosys url-EmployeeSSB
209 209     use_backend prod-EmployeeSSB if src-etsu-wifi-2 host-banner-infosys url-EmployeeSSB
210 210     use_backend prod-EmployeeSSB if src-etsu-sevierville host-banner-infosys url-EmployeeSSB
211 +
212 +     acl url-GeneralSSB path_beg /BannerGeneralSsb
213 +     use_backend prod-GeneralSSB if src-etsu host-banner-infosys url-GeneralSSB
214 +     use_backend prod-GeneralSSB if src-etsu-wifi host-banner-infosys url-GeneralSSB
215 +     use_backend prod-GeneralSSB if src-etsu-wifi-2 host-banner-infosys url-GeneralSSB
216 +     use_backend prod-GeneralSSB if src-etsu-sevierville host-banner-infosys url-GeneralSSB
217
218     acl url-BannerAdminWS path_beg /BannerAdmin.ws
219     use_backend prod-BannerAdmin if src-etsu host-banner-infosys url-BannerAdminWS
...
@@ -368,6 +374,11 @@ backend prod-EmployeeSSB
368 374     balance roundrobin
369 375     server prod-EmployeeSSB-1 palb1.infosys.etsu.edu:80 resolvers ec2-dns check inter 1000
370 376
377 + backend prod-GeneralSSB
378 +     mode http
379 +     balance roundrobin
380 +     server prod-BannerGeneralSsb-1 palb1.infosys.etsu.edu:80 resolvers ec2-dns check inter 1000
381 +
382 backend prod-BannerAdmin
383     mode http
384     balance roundrobin
...
```



# Deployment Demo



# Containerized Banner Apps

- Application Navigator
- Admin Pages
- eTranscript API
- Goldlink (drupal portal)
- College Scheduler
- Banner Online Help
- D2L – LDI
- Banner 8 SSB
- Banner Event Publisher
- RoboRegistrar
- PCI
- SSO Manager
- LYNX (R25)
- Banner 9 Self Service Apps
  - Employee Self Service
  - Student Self Service
  - General Self Service
  - Finance Self Service
  - Banner Extensibility
  - Ellucian Messaging Adapter
  - Integration API
  - Student API
- DegreeWorks Apps
  - Dashboard
  - Scribe
  - Composer
  - 3 API Apps



# Container Examples

- AdminPages
  - tomcat:8-jre7
  - 3 instances
  - 8-12 GB
- Banner 9 Self Service Apps
  - tomcat:8-jre7
  - 2 instances
  - ~2GB
- Banner 8 SSB
  - httpd:2.4 (apache + mod\_owa)
  - 3 instances
  - 100-150MB
- DegreeWorks Apps
  - tomcat:8-jre7 / centos7-java8-sh (composer and tess)
  - 2 instances / 1 instance (composer and admin apps)
  - 1-3 GB



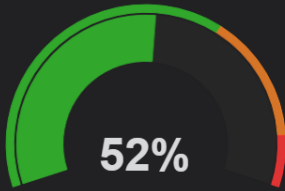
# Monitoring

- Prometheus
- Alerts Manager -> Slack
- Graphana

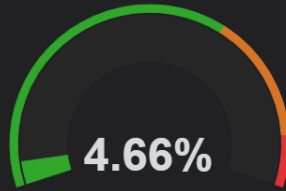


cluster: pcompute container: prod-BannerAdmin

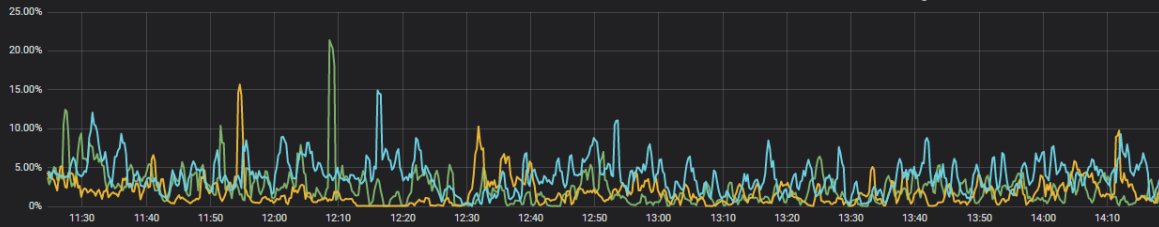
Memory usage (cluster)



CPU usage (cluster)

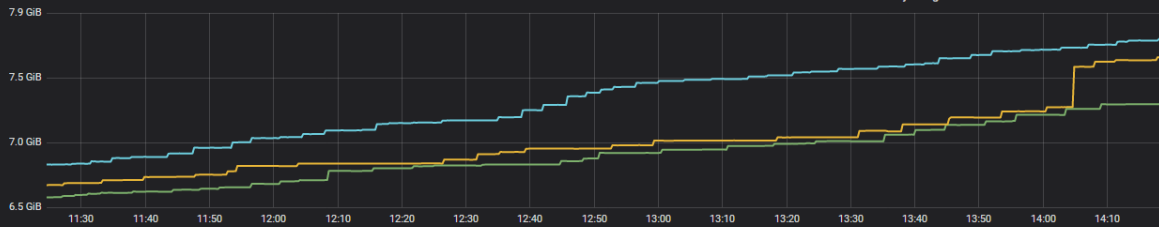


Container CPU usage



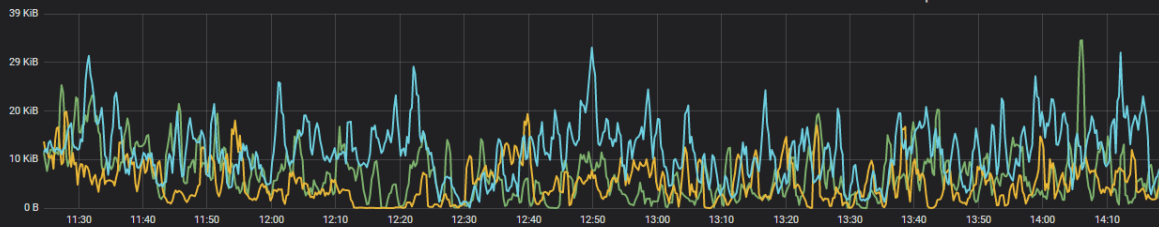
	avg	current
ecs-prod-BannerAdmin-2-prod-BannerAdmin-ccaadce3eccfc6af2800	3.973%	5.380%
ecs-prod-BannerAdmin-2-prod-BannerAdmin-c8a5a392fee8ff1e001	1.824%	3.800%
ecs-prod-BannerAdmin-2-prod-BannerAdmin-9ad18f87bd5a5a62800	2.355%	1.780%

Container Memory Usage



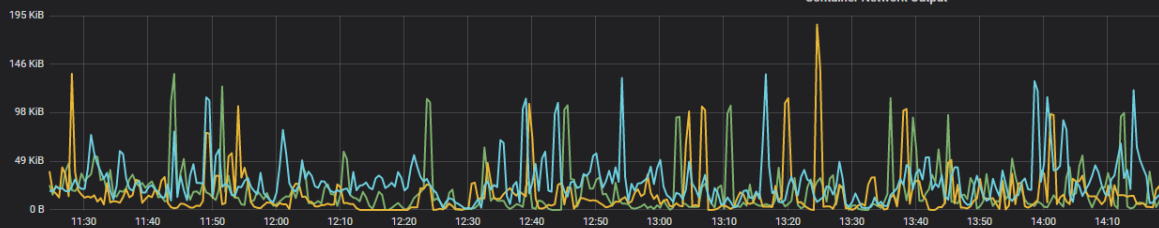
	avg	current
ecs-prod-BannerAdmin-2-prod-BannerAdmin-ccaadce3eccfc6af2800	7.30 GiB	7.73 GiB
ecs-prod-BannerAdmin-2-prod-BannerAdmin-c8a5a392fee8ff1e001	6.99 GiB	7.60 GiB
ecs-prod-BannerAdmin-2-prod-BannerAdmin-9ad18f87bd5a5a62800	6.90 GiB	7.28 GiB

Container Network Input



	avg	current
ecs-prod-BannerAdmin-2-prod-BannerAdmin-c8a5a392fee8ff1e001	5.16 KiB	19.95 KiB
ecs-prod-BannerAdmin-2-prod-BannerAdmin-ccaadce3eccfc6af2800	11.56 KiB	18.51 KiB
ecs-prod-BannerAdmin-2-prod-BannerAdmin-9ad18f87bd5a5a62800	6.80 KiB	5.07 KiB

Container Network Output

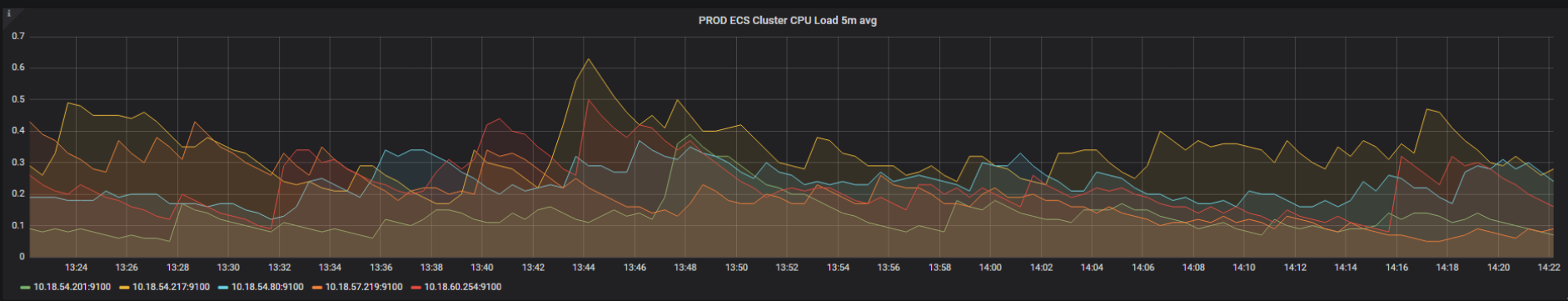
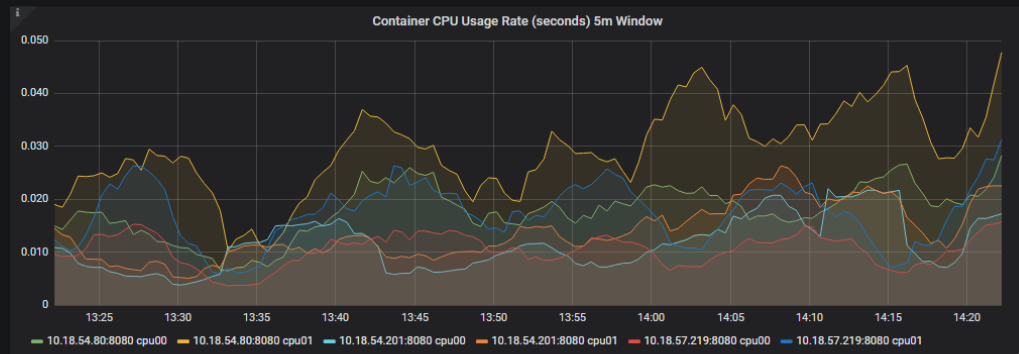
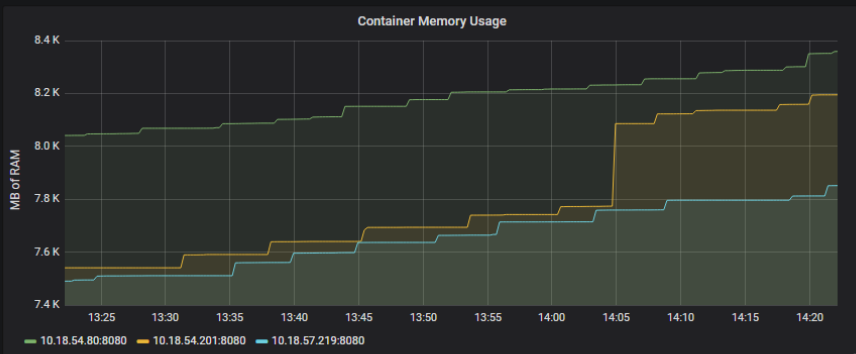
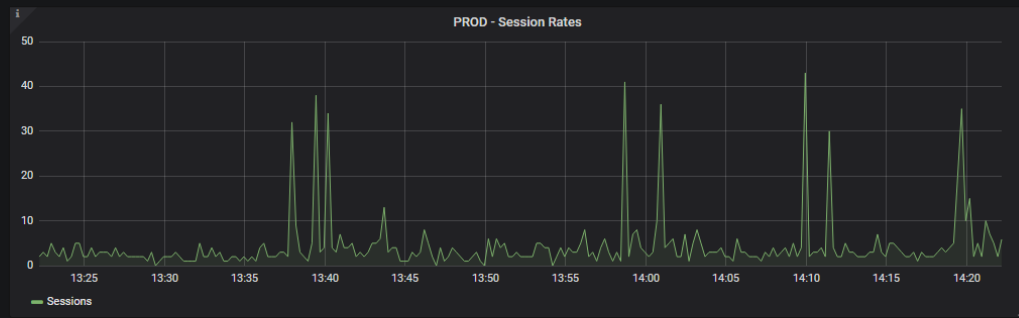
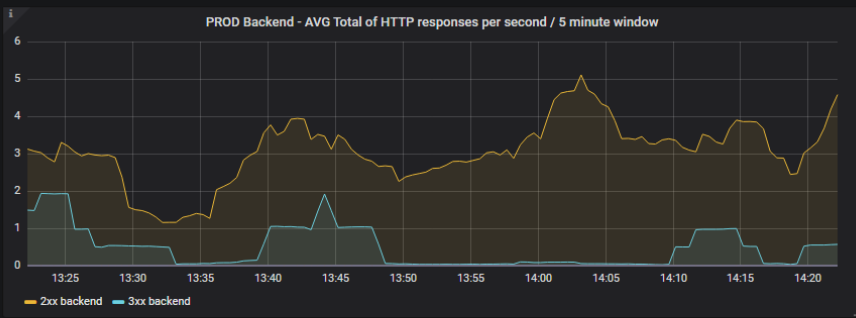


	avg	current
ecs-prod-BannerAdmin-2-prod-BannerAdmin-c8a5a392fee8ff1e001	16.88 KiB	158.59 KiB
ecs-prod-BannerAdmin-2-prod-BannerAdmin-ccaadce3eccfc6af2800	28.64 KiB	33.11 KiB
ecs-prod-BannerAdmin-2-prod-BannerAdmin-9ad18f87bd5a5a62800	19.07 KiB	10.99 KiB



PROD BannerAdmin

Prod (pproxy1) <b>UP</b>	Prod (pproxy2) <b>UP</b>	Recent 5xx Rate <b>0</b>	Backend Current Session Rate /s <b>6</b>	Unique Banproxy Users 24h <b>81</b>	LDI Events Pending <b>3</b>
-----------------------------	-----------------------------	-----------------------------	---	--	--------------------------------



# Docker Logging

- Docker assumes logs are written to STDOUT.
- It's *your responsibility* to do something with log output or log files *inside* your container.
- Logs are ***gone*** when the container is done running / destroyed.



- CloudWatch
- Dashboards
- Alarms
  - ALARM 0
  - INSUFFICIENT 10
  - OK 50
- Billing
- Events
- Rules
- Event Buses
- Logs**
- Metrics
- Favorites
- [+ Add a dashboard](#)

CloudWatch > Log Groups > tsth-BannerAdmin > us-east-1/tsth-BannerAdmin/5750c1ff-1115-4131-986b-b6129f66d450

Time (UTC -04:00)	Message
2018-09-13	13-Sep-2018 13:08:54.751 INFO [main] org.apache.coyote.AbstractProtocol.init Initializing ProtocolHandler ["http-apr-8080"]
▶ 09:08:54	13-Sep-2018 13:08:54.768 INFO [main] org.apache.coyote.AbstractProtocol.init Initializing ProtocolHandler ["ajp-apr-8009"]
▶ 09:08:54	13-Sep-2018 13:08:54.773 INFO [main] org.apache.catalina.startup.Catalina.load Initialization processed in 1229 ms
▶ 09:09:00	13-Sep-2018 13:09:00.083 INFO [main] org.apache.catalina.core.StandardService.startInternal Starting service Catalina
▶ 09:09:00	13-Sep-2018 13:09:00.083 INFO [main] org.apache.catalina.core.StandardEngine.startInternal Starting Servlet Engine: Apache Tomcat/8.0.36
▶ 09:09:00	13-Sep-2018 13:09:00.120 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployWAR Deploying web application archive
▼ 09:09:21	13-Sep-2018 13:09:21.145 INFO [localhost-startStop-1] org.apache.jasper.servlet.TldScanner.scanJars At least one JAR was scanned for TLDs yet contained no TLDs. Some jars failed to parse their manifest - some have been ignored and some have been scanned but no TLDs were found in them. Skipping unneeded JARs during scanning can improve startup time and JSP compilation time.
▶ 09:09:21	configuration: classpath:bannerHelp_configuration.groovy
▶ 09:09:21	configuration: file:/root/.grails/bannerHelp_configuration.groovy
▶ 09:09:21	log4j:WARN No appenders could be found for logger (org.codehaus.groovy.grails.commons.cfg.ConfigurationHelper).
▶ 09:09:21	log4j:WARN Please initialize the log4j system properly.
▶ 09:09:21	log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
▶ 09:09:24	configuration: classpath:bannerHelp_configuration.groovy
▶ 09:09:24	configuration: file:/root/.grails/bannerHelp_configuration.groovy
▶ 09:09:37	13-Sep-2018 13:09:37.308 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployWAR Deployment of web application archive
▶ 09:09:37	13-Sep-2018 13:09:37.315 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployWAR Deploying web application archive
▶ 09:09:37	13-Sep-2018 13:09:37.321 WARNING [localhost-startStop-1] org.apache.catalina.startup.SetContextPropertiesRule.begin [SetContextPropertiesRule]
▶ 09:09:38	13-Sep-2018 13:09:38.036 INFO [localhost-startStop-1] org.apache.jasper.servlet.TldScanner.scanJars At least one JAR was scanned for TLDs yet contained no TLDs. Some jars failed to parse their manifest - some have been ignored and some have been scanned but no TLDs were found in them. Skipping unneeded JARs during scanning can improve startup time and JSP compilation time.
▶ 09:09:38	13-Sep-2018 13:09:38.045 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployWAR Deployment of web application archive
▶ 09:09:38	13-Sep-2018 13:09:38.046 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployWAR Deploying web application archive
▶ 09:10:08	13-Sep-2018 13:10:08.206 INFO [localhost-startStop-1] org.apache.jasper.servlet.TldScanner.scanJars At least one JAR was scanned for TLDs yet contained no TLDs. Some jars failed to parse their manifest - some have been ignored and some have been scanned but no TLDs were found in them. Skipping unneeded JARs during scanning can improve startup time and JSP compilation time.
▶ 09:10:27	13-Sep-2018 13:10:27.993 INFO [localhost-startStop-1] org.apache.catalina.startup.HostConfig.deployWAR Deployment of web application archive
▶ 09:10:28	13-Sep-2018 13:10:28.001 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-apr-8080"]
▶ 09:10:28	13-Sep-2018 13:10:28.017 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["ajp-apr-8009"]
▶ 09:10:28	13-Sep-2018 13:10:28.028 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in 93254 ms



# Start with Tomcat

- Doesn't rely on file-system persistence
- Horizontal scaling
- Common communication protocol: HTTP
  
- Better ROI on technical investment
  - More individual Middleware deployments
  - High overlap of implementation details among tomcat deployments
  
- Solved problem
  - Plenty of examples and support



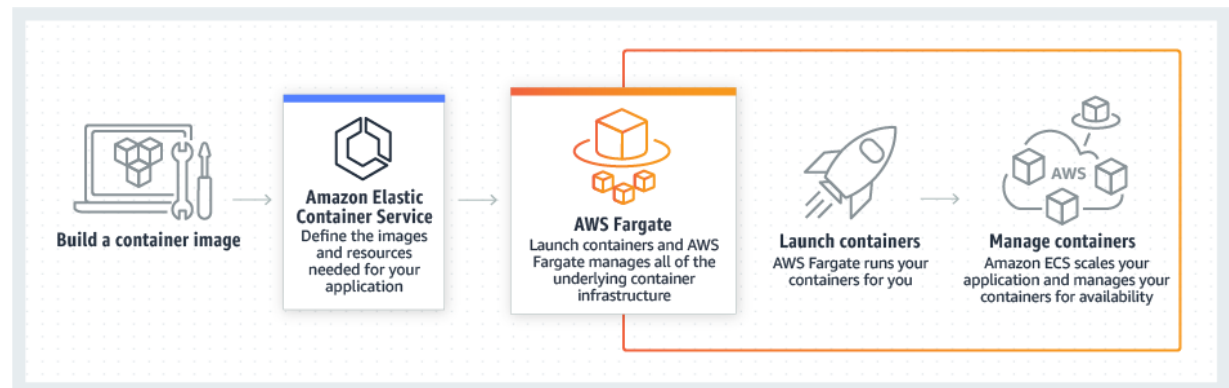
# Summary

- Get comfortable with Docker
  - Cattle not pets
- Kubernetes or ECS
  - Shift & Lift or Lift & Shift
- Baked images then Base images



# Whats Next?

- Containerize more apps
  - EIS
  - Rabbitmq
  - IFEP
  - Jobsub
  - UC4
- Increase usage of CI/CD
- Increase monitoring via Prometheus and alerting
- Greenfield
  - AWS Fargate



# Questions?



# Resources

- <http://www.gabrieltozzi.com/talks>
- BanDock: Docker Group
  - [bandock@googlegroups.com](mailto:bandock@googlegroups.com)
  - <https://bitbucket.org/edurepo/>

