

Running Your Services On Docker

An experience report

Who Am I?

Robert Bastian

Director, Platform and Architecture at Drillinginfo

20+ years industry experience in Telcos, Gaming
and Energy

I love APIs and services!

Agile and DevOps advocate



Why Docker?

My World Needed To Change

5+ individual teams building “micro services” in Java and Scala

Frictionless deployment of “micro-services” using Chef & AWS

25+ separate “micro-services” deployed in the previous 18 months

Each service is typically deployed to a single AWS virtual machine

Each service is deployed 6x - dev, test, staging (2x) and production (2x)

25+ “micro-services” became nearly 150 AWS virtual machines

Why Docker? COST!

The AWS bill is too damn high!

Decline in the global price of oil causing churn in our business

6 AWS virtual machines per service isn't sustainable with our budget

AWS monthly bill started to gain visibility from sr. management and ***the board***

Why Docker? WASTE!

We weren't using the compute and memory resources purchased from AMZN!

Nearly all “micro-services” were at 1% CPU utilization

Nearly all “micro-services” were only using 40% of memory (JVM)

150+ virtual machines essentially sitting idle

Why Docker? LOCK IN!

How would we leave AMZN if we wanted to?

Could we use Drillinginfo IT's Openstack platform?

What about alternate IaaS providers like Rackspace or Azure?

What about Container as a Service (CaaS) providers like Joyent, Tutum or Profitbricks?

What about using Amazon's Container Service?














My World Needs To Change - Problem Statement

“How can we *deploy fewer* virtual machines while *increasing the density and utilization* of services per machine *without locking* us into a specific IaaS provider?”

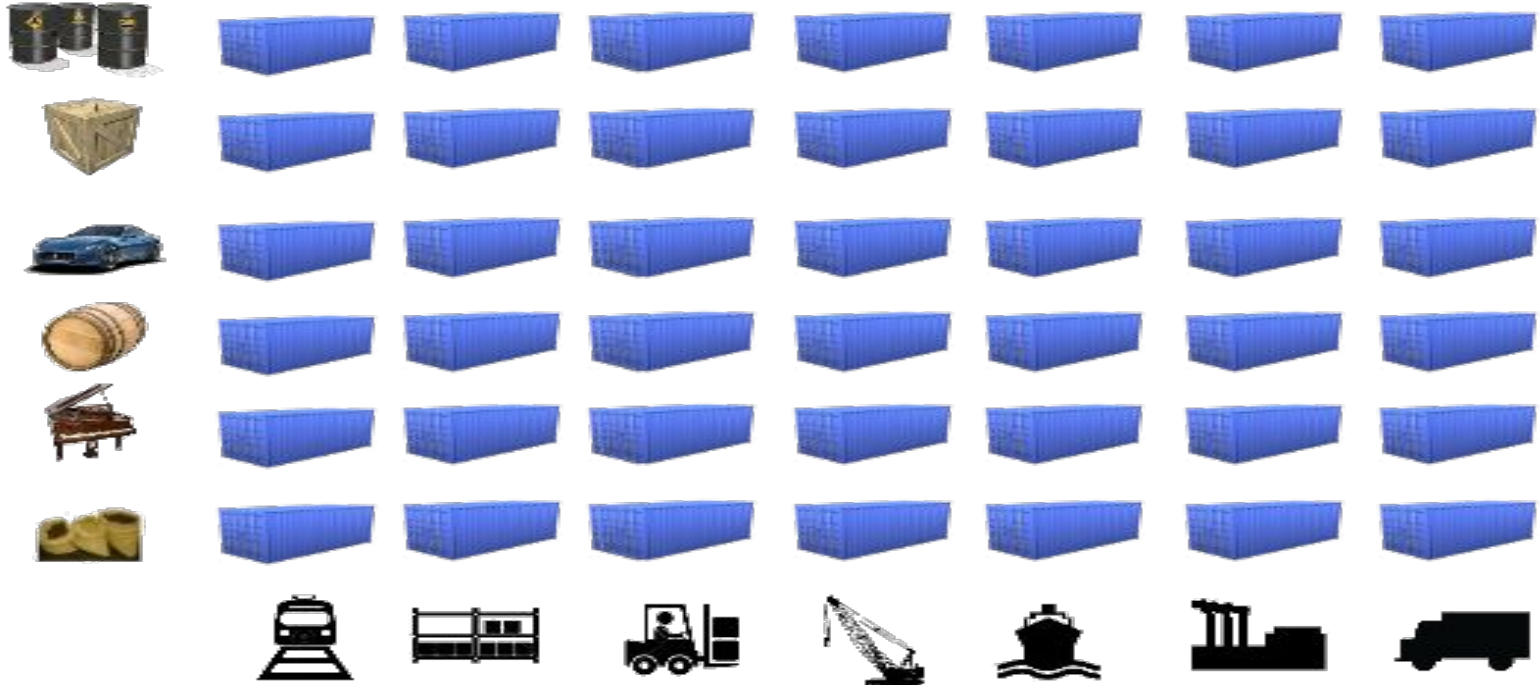


How Docker Solves All The Problems

Docker Containers - Shipping Matrix From Hell

	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
	?	?	?	?	?	?	?
							

Docker Containers - Standard Shipping Container

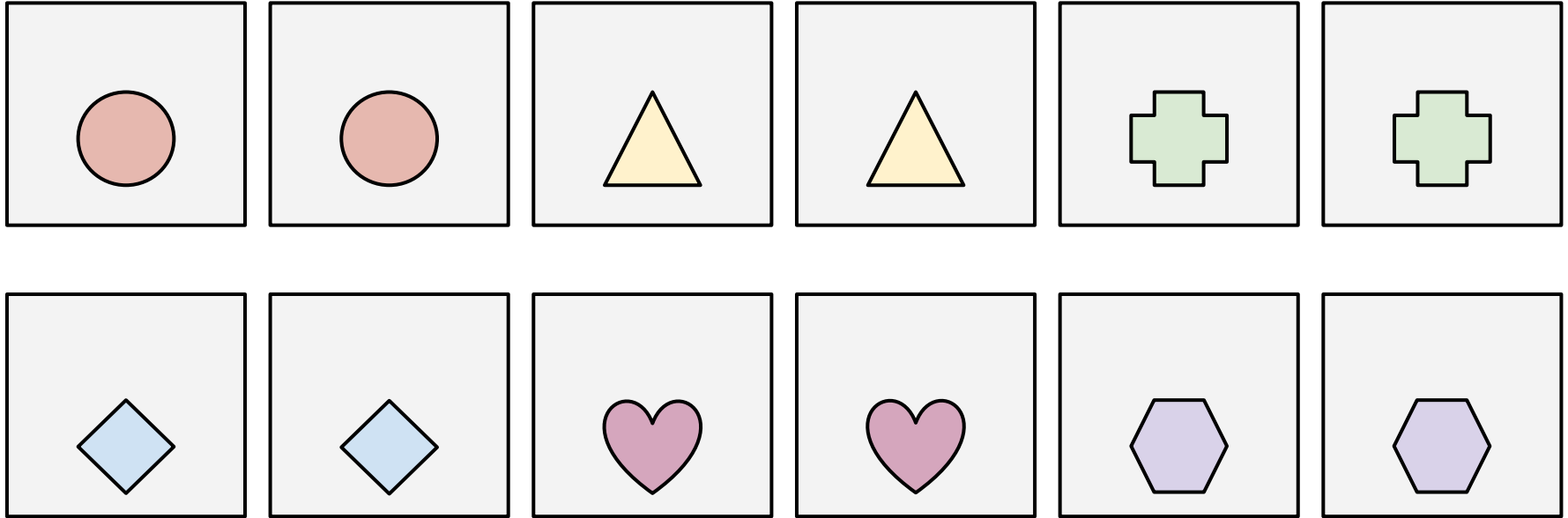


What's Inside Doesn't Matter



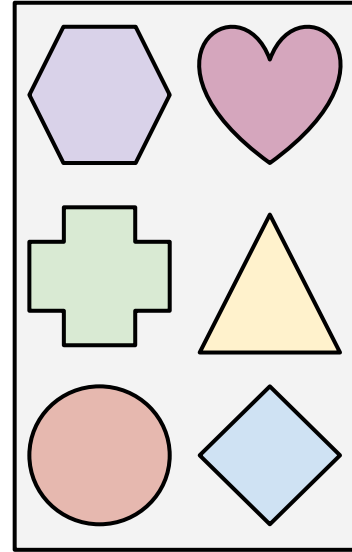
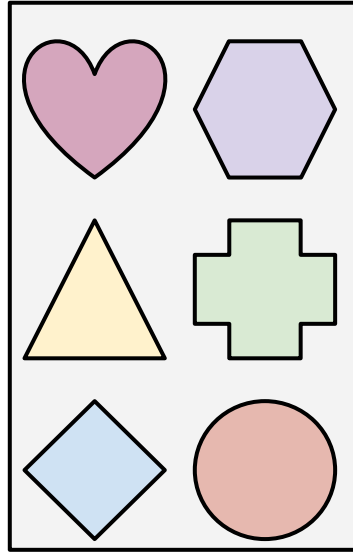
Why Docker Is Important - Before Containers

Very inefficient use of memory and CPU resources



Why Docker Is Important - After Containers

Isolated
services in
fewer
VMs...



... and use
VMs more
efficiently.

Why Is Docker Important?

Docker container technology provides our “micro-services” platform:

Increased ***density*** of ***isolated*** “micro-services” per virtual machine (9:1!)

Containerized “micro-services” are ***portable*** across machines and providers

Containerized “micro-services” are much ***faster*** than virtual machines



Containers Alone Aren't Enough

But Containers Aren't Enough!

Running containerized “micro-services” in production requires ***much more*** than just Docker.

It requires a “Platform” that can do the following:

- Building and pushing Docker images to an image repository

- Pulling images, provisioning and scheduling containers

- Discovering and binding to services running as containers

- Containers discovering and binding to other containers

- Operating and managing services in containers

Drillinginfo Docker Platform: Build & Store Images



Drillinginfo Docker Platform: Jenkins & Dockerhub

Problem: How do we **build** images? Jenkins automates the image builds.

We started building our images with Ubuntu 14.04 (1GB)

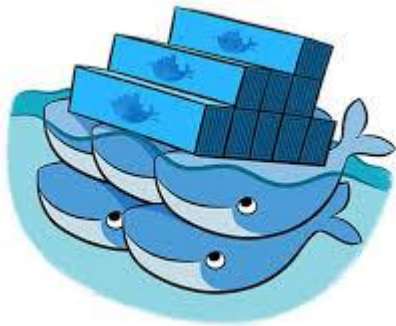
We settled on **Alpine**, a minimal linux distribution (5MB)

Typical “micro-services” now ~ 390MB

Problem: Where do we **put** them? Dockerhub.

- Tried Docker Trusted Registry and Core OS Enterprise Registry
- Settled on using **Dockerhub**
- Use **latest and sem-ver** tags on our images

Drillinginfo Docker Platform: Provisioning, Scheduling



Dockerhub



Drillinginfo Docker Platform - Chef

Problem: How do we determine ***which host*** to run a container on and how do we ***configure and start*** the container?

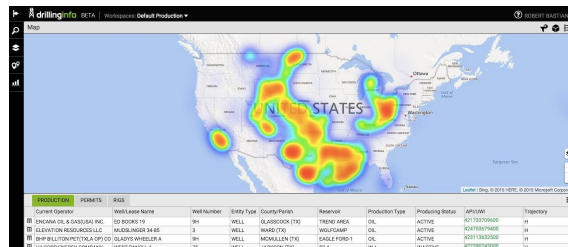
We solve ***scheduling and provisioning*** with Chef.

Chef ***schedules*** containers on specific hosts using ***Chef roles***

Chef ***provisions*** and ***configures*** containers using Chef ***recipes and environments***

Each “micro-service” has an associated Chef ***recipe*** that converts Chef attributes into ***container environment variables***

Drillinginfo Docker Platform: Service Directory

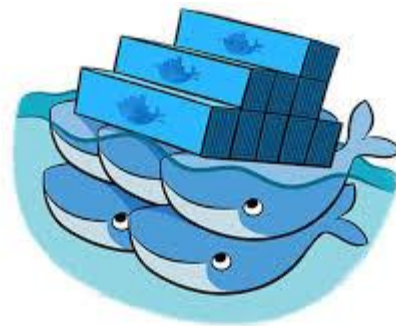


DI Web
Applications

Problem:
How can web



bind to
containers?



DI Docker
Containers

Drillinginfo Docker Platform - Consul

Problem: How do our browser applications *locate* service containers?

We use Hashicorp's Consul as our *service directory*.

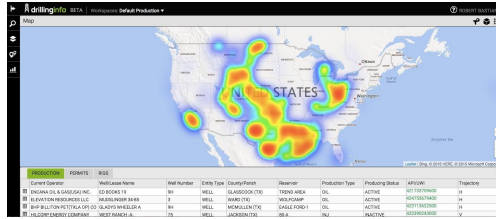
Containers *automatically register themselves* with Consul when started.

The Docker daemon *emits real-time lifecycle events* for container start

We use a utility container called *Registrar* to automate the registration of “micro-service” containers with Consul

Containers are registered with a *health check* that Consul polls to determine the health of the container

Drillinginfo Docker Platform: Service Discovery



Problem: How can web applications *discover* and *bind* to containers?



CONSUL
TEMPLATE



Drillinginfo Docker Platform - Consul Template

Problem: How do our browser applications use services deployed in containers?

We use Hashicorp's Consul Template for service discovery and Varnish for load balancing.

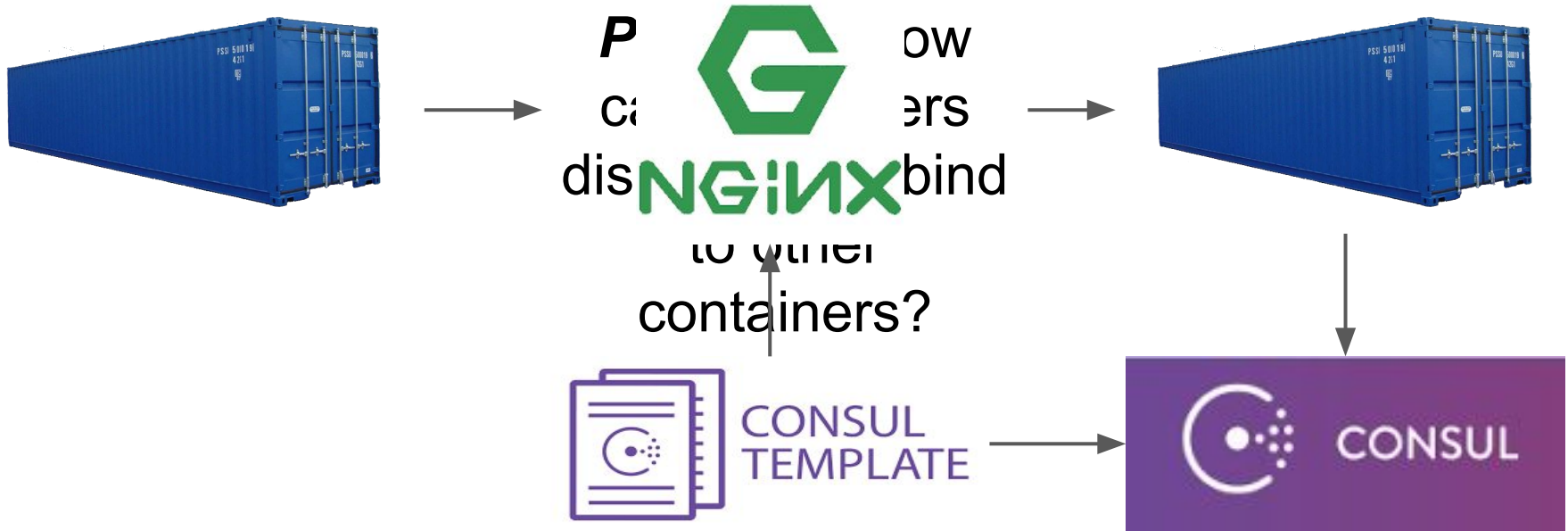
Consul Template detects containers in Consul and updates Varnish configuration

Consul Template participates in the Consul cluster using Consul Client

Consul Template automatically **adds healthy** containers and **removes sick** containers from the Varnish load balancer by updating Varnish configuration

Browser applications use **Varnish routes** to reach services running in containers

Drillinginfo Docker Platform: Container Dependencies



Drillinginfo Docker Platform - Service Proxy

Problem: How can containers find their containerized dependencies on the same host and different hosts?

We use Consul, Nginx and Consul Template to implement a “Service Proxy” for inter and intra-host container communication.

- We built a utility container called “Service Proxy” that uses Consul’s service directory to locate a container’s ip address and port
- “Service Proxy” then uses Consul Template to create an nginx.conf with load balanced routes for each service container
- ***Docker Links*** work for intra-host dependencies but with a ***gotcha***

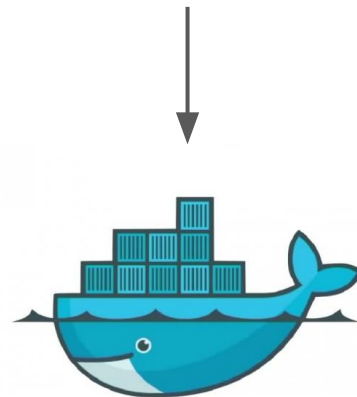
Drillinginfo Docker Platform: Operations & Monitoring



Uptime Events Checks Tags Up 38

Tags last 24 hours

Name	Availability	Uptime	Response Time
arcgis-map-service	100%	<div></div>	18ms
atlas-service	100%	<div></div>	17ms
auth-service	100%	<div></div>	13ms
completions-tile-service	100%	<div></div>	14ms
entitlements-service	100%	<div></div>	15ms
exports-service	100%	<div></div>	14ms
feature-gates-service	100%	<div></div>	17ms
identity-service	100%	<div></div>	19ms
image/drillinginfo/entitlements-service:2.21.4	100%	<div></div>	15ms
ins-file-download-service	100%	<div></div>	15ms
license-generator-service	100%	<div></div>	22ms
mobile-subscriptions-service	100%	<div></div>	56ms
papi-service	100%	<div></div>	56ms



Drillinginfo Docker Platform - Operations & Monitoring

Problem: How do we ***monitor containers*** and ***notify and escalate*** when containerized services aren't healthy?

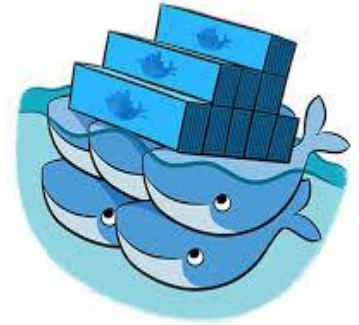
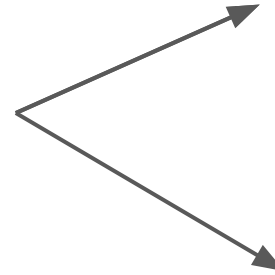
We use Uptime and VictorOps monitor our containerized services.

- A utility container monitors Docker container lifecycle events and ***automatically registers a service check with Uptime*** when a container starts
- Uptime ***service interruptions to*** VictorOps for on-call scheduling, paging and escalation

Drillinginfo Docker Platform: Operations & Monitoring

Pro / do
we re ne
resc e of

DATADOG
containers?



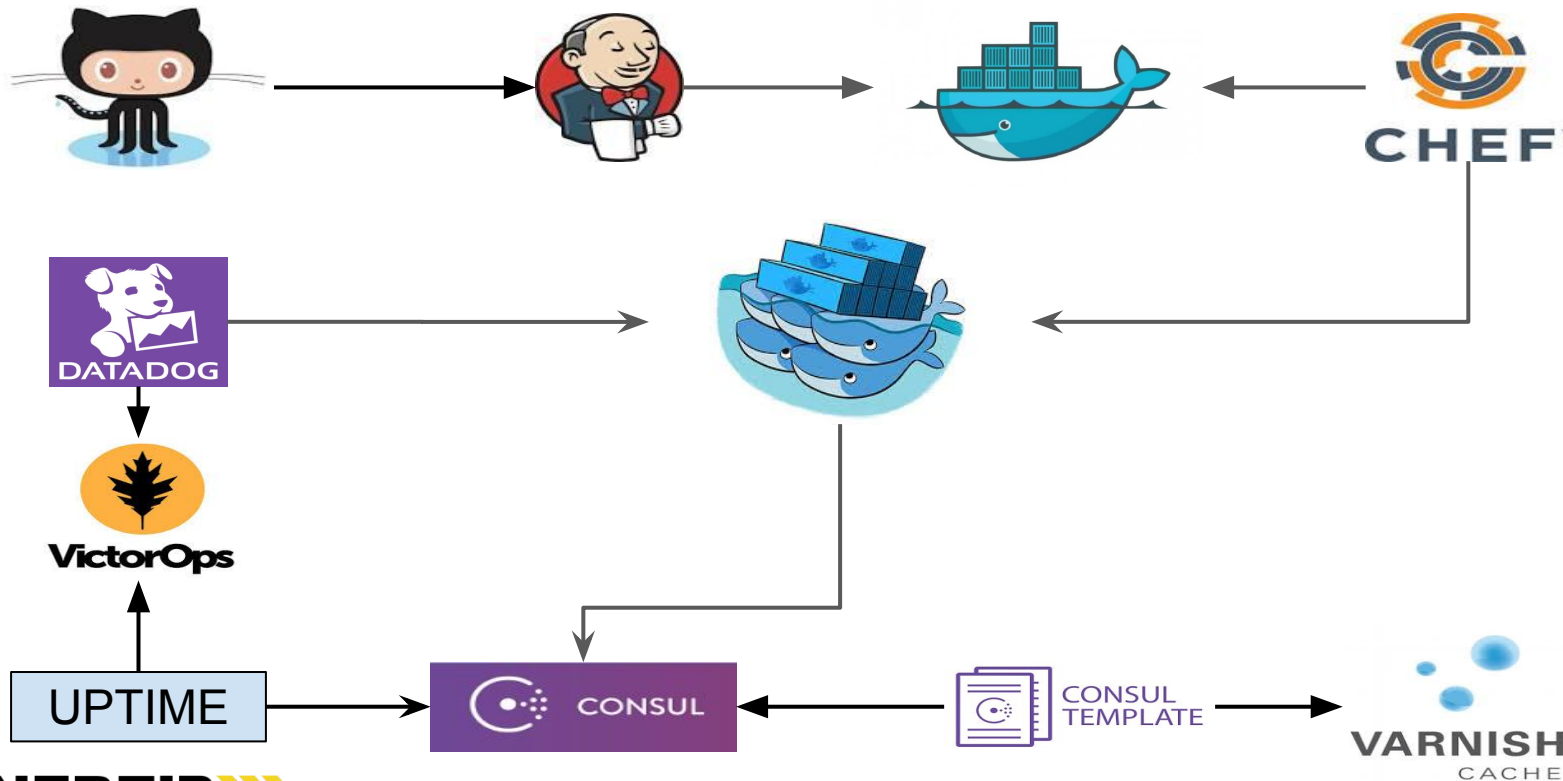
Drillinginfo Docker Platform - Operations & Monitoring

Problem: How do we **monitor** our Docker host's **resource usage**?

We use Datadog to monitor the Docker host utilization and the service's metrics.

- Datadog helps us visualize the resource usage on a host
- Datadog helps us understand how our services are performing
- Datadog helps us understand how to “pack” containers onto hosts by exposing the current utilization of CPU and memory resources on the host

Drillinginfo Docker Platform - Overview



Drillinginfo Docker Platform - Wrap Up

The Docker container technology and the Drillinginfo Docker Platform provide our “micro-services” infrastructure the following benefits:

- Reduced cost for IaaS hosting

- Reduced waste of virtual machine resources

- Standardized deployment mechanism for “micro-services”

- Standardized service directory, service discovery

- Standardized metrics dashboards, monitoring and alerting

Drillinginfo Docker Platform - Future

Chef has gotten us where we are today but ***not where we want to be.***

Container orchestration

Host provisioning and pooling

Drillinginfo Docker Platform - Orchestration

Docker Compose will **replace Chef roles** defining the “micro-services” deployed on our platform and which Docker host they run on.

The Docker Compose YAML file:

- Defines which containerized “micro-services” run on which host

- Define the environment variables for each container

I believe that IaaS providers will standardize on Docker Compose for container orchestration.

Drillinginfo Docker Platform - Provisioning & Pooling

Docker Machine will ***replace Chef for provisioning virtual machines*** with Docker.

Docker Machine automates the provisioning of Docker hosts

Docker Swarm will ***replace Chef for scheduling containers*** on a host.

Swarm combines Docker Machines into a single pool of compute and memory resources

Swarm provides container scheduling and supports plug-in schedulers

Docker Compose will define all the containers that run on the Swarm

Running Your Services On Docker: Thank You!

Questions?

Contact Info

Please feel free to contact me with any additional questions or comments!

Email: robert.bastian@drillinginfo.com

LinkedIn: rbastian

Twitter: @rbastian

Running Your Services On Docker - Links

<https://www.docker.com/>

<https://hub.docker.com/>

<https://jenkins-ci.org/>

<https://www.chef.io>

<https://www.consul.io/>

<https://github.com/gliderlabs/registrator>

<https://hashicorp.com/blog/introducing-consul-template.html>

<https://www.varnish-cache.org/>

<https://www.nginx.com/>

<https://github.com/fzaninotto/uptime>

<https://victorops.com/>

<https://www.datadoghq.com/>



www.synerzip.com

Ashish Shanker

Ashish.Shanker@synerzip.com

469.374.0500

Synerzip in a Nutshell

- ◆ Software product development **partner** for small/mid-sized technology companies
 - Exclusive focus on small/mid-sized technology companies, typically venture-backed companies in growth phase
 - By definition, all Synerzip work is the IP of its respective clients
 - Deep experience in full SDLC – design, dev, QA/testing, deployment
- ◆ Dedicated team of **high caliber** software professionals for each client
 - Seamlessly extends client's local team offering full transparency
 - Stable teams with very low turn-over
 - NOT just “staff augmentation, but provide full management support
- ◆ Actually **reduces risk** of development/delivery
 - Experienced team – uses appropriate level of engineering discipline
 - Practices Agile development – responsive yet disciplined
- ◆ **Reduces cost** – dual-site team, 50% cost advantage
- ◆ Offers long-term **flexibility** – allows (facilitates) taking offshore team captive – aka “BOT” option

Synerzip Clients



Next Webinar

Role of the Architect in Agile



**Complimentary Webinar:
Thursday, November 12, 2015
@ Noon CST**

Presented by: Chris Edwards, P.Eng
Software Manager, IHS Inc.

Connect with Synerzip



**@Syner
zip**



**linkedin.com/company/syner
zip**



**facebook.com/Syner
zip**

Ashish Shanker

Ashish.Shanker@synerzip.com

469.374.0500

Running Your Services On Docker: Thank You!

Questions?