Using Elasticsearch to Help Monitor Docker Containers

OCTOBER 19, 2016

Tonight's Discussion

- Introduction into the Elastic Stack and Beats
- Overview of tonight's demonstration setup
- Topbeat configuration
- Assembling the Elastic Stack
- Assembling the sample application
- Interspersed with Docker 1.12 Swarm and Routing Mesh

Tonight's Discussion

- We'll be moving pretty quickly
  - This presentation is targeted toward an intermediate level
    - Some a bit more advanced level
  - Due to time, not much looking into the Dockerfiles
    - Happy to discuss after the presentation
    - They will be made available
- Do ask questions
  - Some may be deferred to the end of the session
  - There will be a Q&A at the end

All Artifacts will be Available on GitHub

In few days; maybe early next week.
A note will be posted on the Meetup site as soon as they are available.

The Elastic Stack

The Elastic Stack

Elasticsearch 2.4.1
a distributed, open source search and analytics engine

Kibana 4.6.1
an open source data visualization platform that allows you to interact with Elasticsearch data

Beats 1.3.1
open source data shippers for Elasticsearch

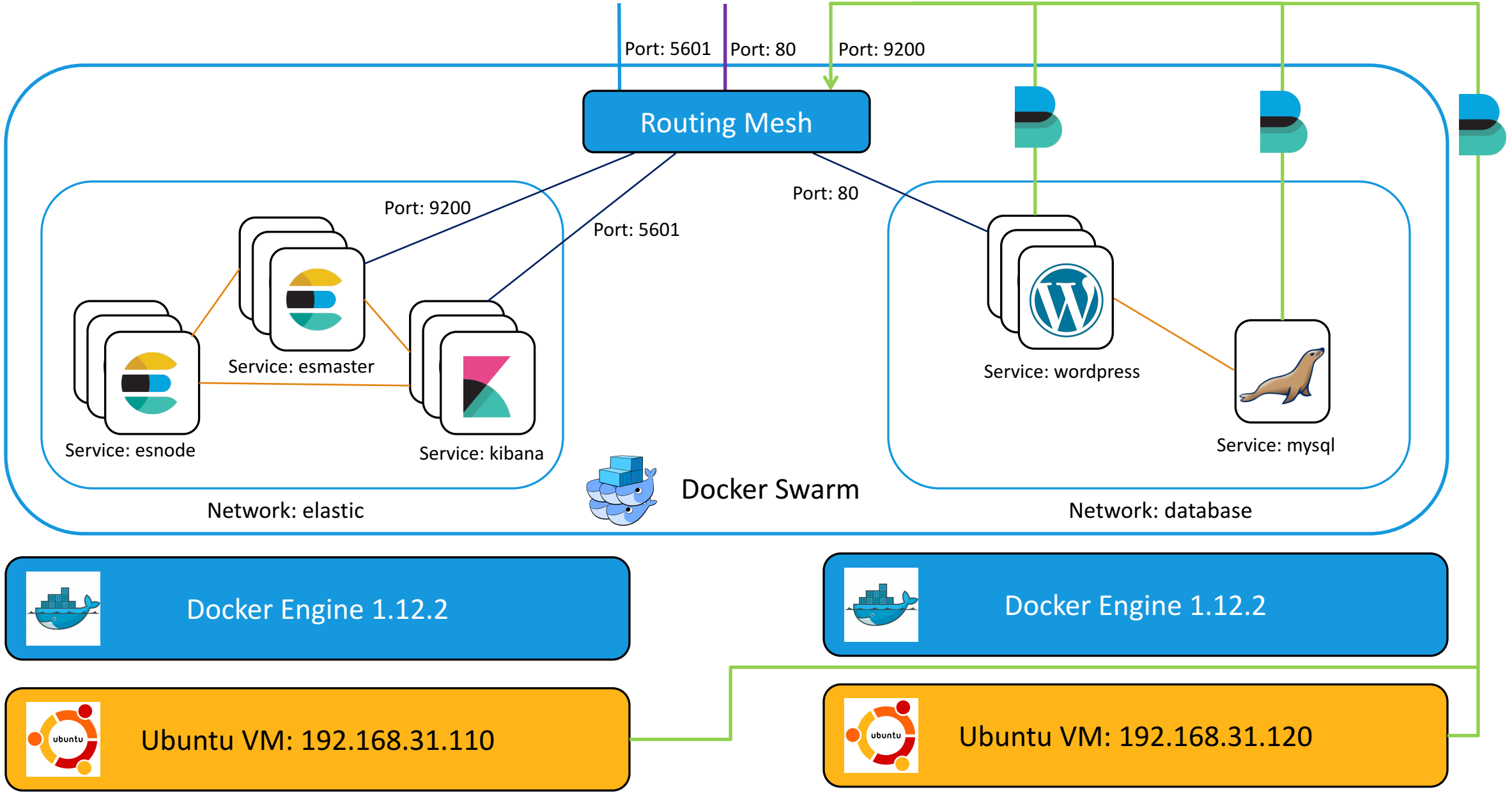
Marvel 2.4.0
provides status for an Elasticsearch deployment

Soon to be version 5.0 across the suite

Shipping Data to Elasticsearch

- Filebeat: Analyze Log Files in Real Time
- Packetbeat: Tap into Your Wire Data
- Winlogbeat: Gather Insight from Windows Event Logs
- Topbeat: Gather Infrastructure Metrics
- Libbeat: For Building More Beats
- Several community contributions as well

Demonstration Setup



Topbeat Configuration

Capturing System and Process Data

- System-wide
  - CPU, memory, swap, etc.
- Per-process
  - Name, PID, state, CPU, memory
- File system
  - Disk mount information, available space

Straightforward Configuration

topbeat.yml

```
input:
  # In seconds, defines how often to read server statistics
  period: 10

  # Regular expression to match the processes that are monitored
  # By default, all the processes are monitored
  procs: [".*"]

  # Statistics to collect (all enabled by default)
  stats:
    # per system statistics, by default is true
    system: true

    # per process statistics, by default is true
    process: true

    # file system information, by default is true
    filesystem: true

    # cpu usage per core, by default is false
    cpu_per_core: false
```

Two Configurations for the Demo

- Topbeat on the VMs
  - VM system-wide data
  - Without per-process data

```
# per system statistics, by default is true
system: true

# per process statistics, by default is true
process: false
```

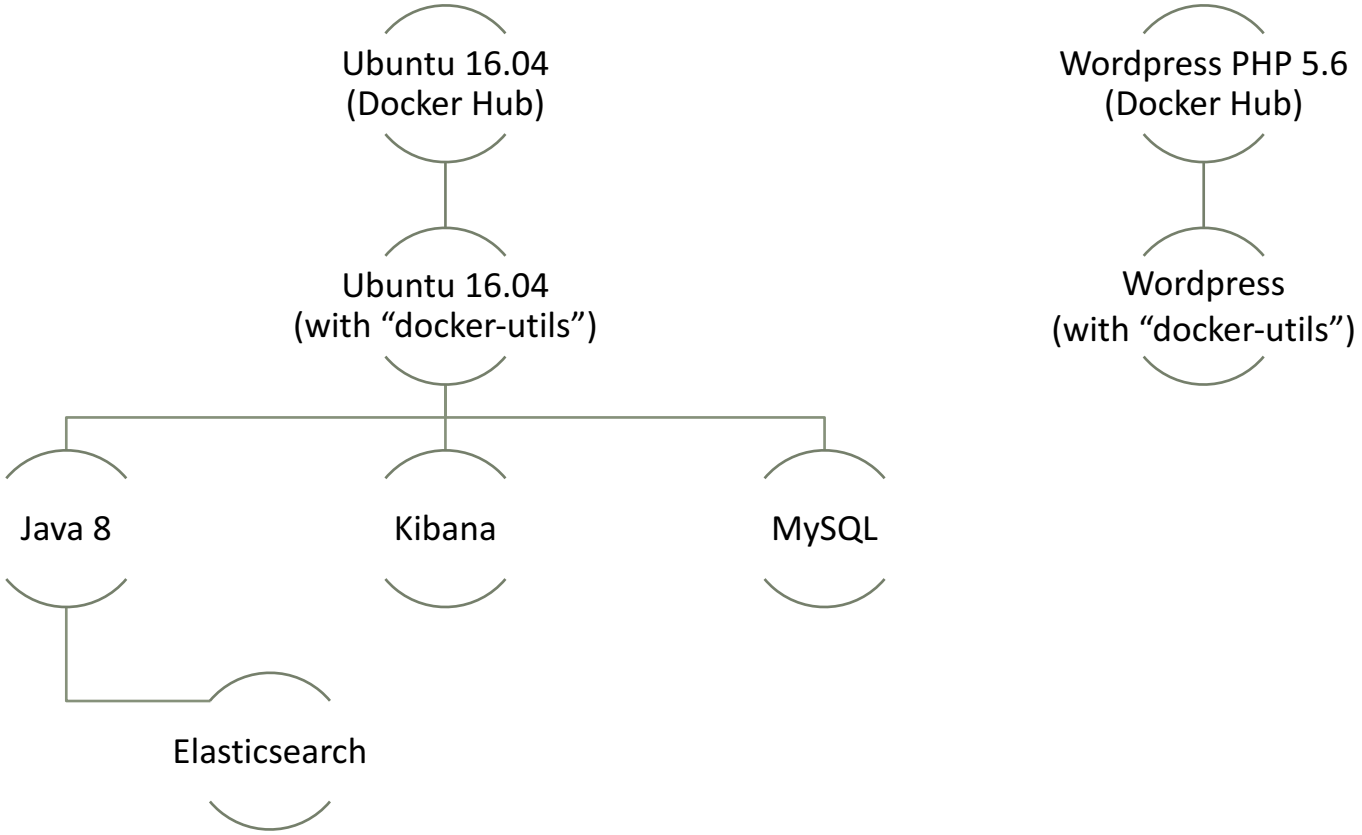
- Topbeat in the containers
  - Per-process data
  - Without system-wide data

```
# per system statistics, by default is true
system: false

# per process statistics, by default is true
process: true
```

Image Hierarchy

Image Hierarchy



Assembling the Elastic Stack

Elasticsearch Master Node Service

```
docker service create \
    --replicas 3 \
    --publish 9200:9200 \
    --name esmaster \
    --network elastic \
    --mount type=bind,source=/opt/esdata,destination=/home/elastic/data \
    meetup/elasticsearch:2.4.1 \
    /home/elastic/bin/start-elasticsearch.sh -m 2
```

Let's Take a Look

Kibana Service

```
docker service create \
    --publish 5601:5601 \
    --name kibana \
    --network elastic \
    meetup/kibana:4.6.1 \
    /home/kibana/bin/start-kibana.sh -s esmaster
```

Let's Take a Look

Elasticsearch Worker Node Service

```
docker service create \
    --replicas 2 \
    --name esnode \
    --network elastic \
    --mount type=bind,source=/opt/esdata,destination=/home/elastic/data \
    meetup/elasticsearch:2.4.1 \
    /home/elastic/bin/start-elasticsearch.sh -m 2
```

Let's Take a Look

Assembling the Sample Application

MySQL Service

```
docker service create \
     --name mysql \
     --network database \
     --env ES_SERVER=192.168.31.110 \
     --env MYSQL_ROOT_PASSWORD=mysqlroot \
     --constraint engine.labels.meetup.node==mysql \
     meetup/mysql:5.7
```

Let's Take a Look

Wordpress Service

```
docker service create \
    --name wordpress \
    --network database \
    --publish 80:80 \
    --env ES_SERVER=192.168.31.110 \
    --env WORDPRESS_DB_PASSWORD=mysqlroot \
    --mount type=bind,source=/opt/wordpress,destination=/var/www/html \
    --constraint engine.labels.meetup.node==wordpress \
    meetup/wordpress:php5.6
```

Let's Take a Look

Questions?

Using Elasticsearch to Help Monitor Docker Containers

OCTOBER 19, 2016