# Docker Kubernetes Support

What does it mean for you?



# The Big Announcement

- In October 2017, Docker broke the big news that they'll be adding Kubernetes support to Docker:

  https://blog.docker.com/2017/10/kubernetes-docker-platform-and-moby-project/

- It will work by putting Kubernetes and Swarm on equal footing as orchestrator choices for Docker

# Docker: Now Powered by Swarm and Kubernetes


(Source: https://www.docker.com/kubernetes)

docker

controlplane

Amazic

# The Big Announcement

- What progress has been made since the announcement?
- When can we expect it to be ready?
- How does it work?

*This is what I'll be talking about today!*

**Disclaimer:** I don't work for Docker and don't know their strategy or roadmap beyond what has been publicly stated

# The Announcements in More Detail

- From Solomon's announcement blog post:

  *"...check out the detailed blog posts to learn how we're bringing Kubernetes to:*

  - *Docker Enterprise Edition*
  - *Docker Community Edition on the desktop with Docker for Mac and Windows*
  - *The Moby Project"*
- And from https://www.docker.com/kubernetes:

  *"We're adding Kubernetes support in both Docker Community Edition on the desktop for developers using Windows and macOS, and in Docker Enterprise Edition. The bulk of the Kubernetes integration work happens in the open in the Moby Project.*

# The Announcements in More Detail

- Note that desktop Docker CE is the only Docker CE getting a mention; they're very specific about where it's supported in Docker CE
- Clearly Docker are focusing their investment on a workflow from desktop to Docker EE in production, therefore Docker CE Linux server use-case is seemingly not on the roadmap

# What is Available to us Today?

- The announcement said that most of the integration work happens in Moby
- In part, that's referring to LinuxKit, a tool for building minimal Linux OS images, described in YAML
- It underpins Docker for Mac and Docker for Windows
- Today, you can use the Docker for Mac or Docker for Windows installers to get Kubernetes on a desktop that you can deploy to with Docker

# What is Available to us Today?

- This requires Docker for Mac **17.12 CE Edge** and higher or Docker for Windows **18.02 CE Edge**. Docker EE Kubernetes support requires access to the beta programme
- Here are the links to the announcements:
  - Mac, 09/01/2018: https://blog.docker.com/2018/01/docker-mac-kubernetes/
  - Win, 30/01/2018: https://blog.docker.com/2018/01/docker-windows-desktop-now-kubernetes/
  - Docker EE, 18/01/2018: https://blog.docker.com/2018/01/docker-ee-kubernetes/
- To play with Kubernetes on Docker, try Docker for Mac & Windows desktop

# I’m a Developer; what can I do with it?

The following example is taken from the Docker Blog, here:

https://blog.docker.com/2018/02/docker-compose-kubernetes-docker-desktop/

- Deploy a compose stack to Kubernetes:

```
$ docker stack deploy --compose-file words.yaml words
Stack words was created
Waiting for the stack to be stable and running...
 - Service db has one container running
 - Service words has one container running
 - Service web has one container running
Stack words is stable and running
```

# I’m a Developer; what can I do with it?

- This has created Kubernetes resources under the hood:

```
$ kubectl get deployment
NAME      DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
db        1         1         1            1           2m
web       1         1         1            1           2m
words     5         5         5            5           2m

$ kubectl get pods
NAME                     READY     STATUS    RESTARTS   AGE
db-5489494997-2krr2      1/1       Running   0          2m
web-dd5755876-dhnkh      1/1       Running   0          2m
words-86645d96b7-8whpw   1/1       Running   0          2m
words-86645d96b7-dqwxp   1/1       Running   0          2m
words-86645d96b7-nxgbb   1/1       Running   0          2m
words-86645d96b7-p5qxh   1/1       Running   0          2m
words-86645d96b7-vs8x5   1/1       Running   0          2m

$ kubectl get services
NAME            TYPE           CLUSTER-IP       EXTERNAL-IP    PORT(S)       AGE
db              ClusterIP      None                            55555/TCP     2m
web             ClusterIP      None                            55555/TCP     2m
web-published   LoadBalancer   10.104.198.84                   80:32315/TCP  2m
words           ClusterIP      None                            55555/TCP     2m
```

# I’m a Developer; what can I do with it?

- We can see here some artefacts of Docker’s translation from a Compose stack into Kubernetes services
- The single `web` Docker service with an exposed port becomes two Kubernetes services- `web` and `web-exposed`; the latter of which is a load balancer

You can watch a video of Elton Stoneman explaining all of this here:

https://www.youtube.com/watch?time_continue=2&v=h4J8xQWlsQw

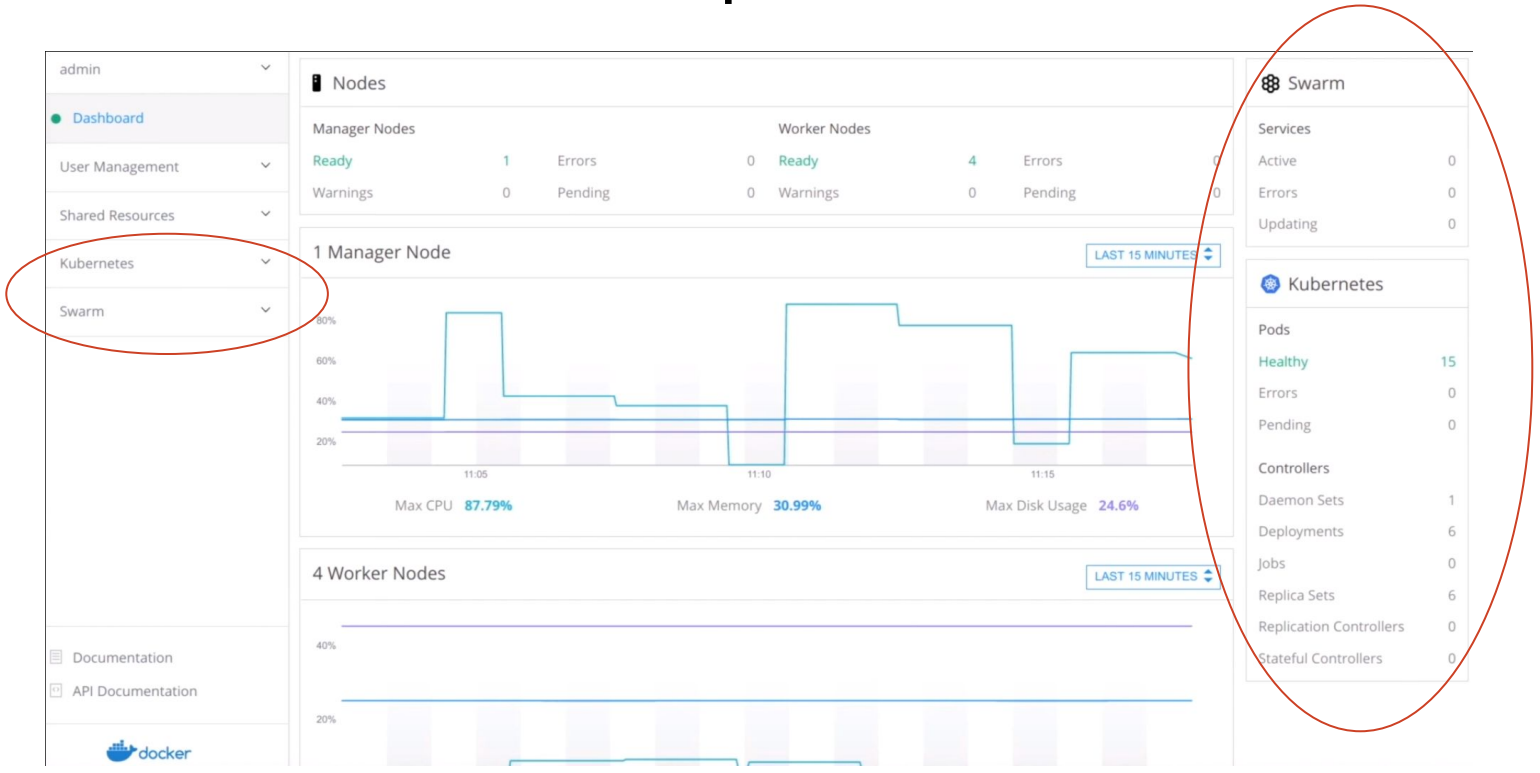
# What about Docker Enterprise Edition?

- As you have seen, it's clear that at least the basic Docker Kubernetes integrations are in place, even if a little bleeding-edge
- What about Docker EE?
- As you no doubt know, Docker EE consists of Docker Datacenter (Enterprise Docker distribution) and Universal Control Plane (its UI dashboard)
- There is a video showing UCP with Kubernetes support here:

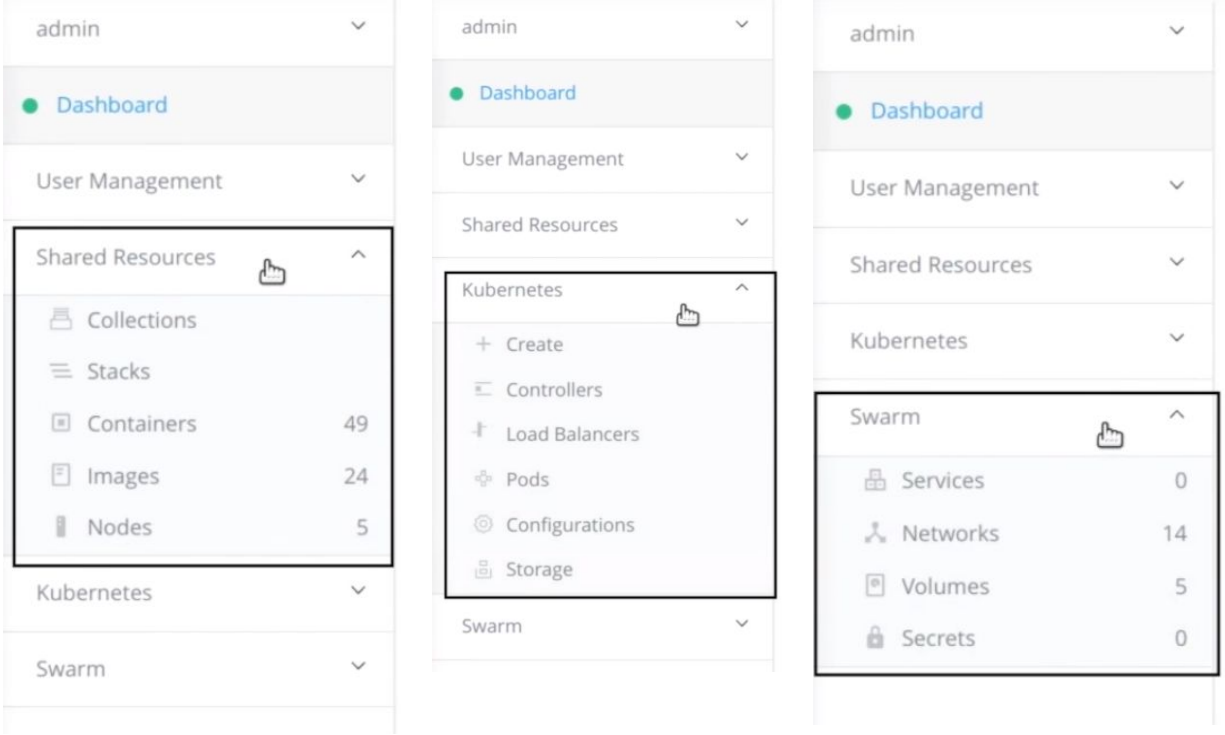
https://youtu.be/h2B6mhDCw2o

- I have some screenshots from it

# What about Docker Enterprise Edition?



# What about Docker Enterprise Edition?




docker

controlplane

Amazic

# What about Docker Enterprise Edition?

- Docker have some new features that arrive with the Kubernetes integration:
- Kubernetes integration to Docker EE access controls
  - One of the biggest features Enterprise needs is good RBAC; Docker EE has this, and they've extended it to Kubernetes
- Kubernetes integration to Docker Trusted Registry
  - Controls around the Docker EE registry for signing and image promotions apply across orchestrators

# So what does this mean for you?

- If you're a Docker shop, you won't be left behind by Kubernetes taking over
- If you found Swarm to be limiting, you now have the full power of Kubernetes
- It looks like you have the best of both worlds
- It doesn't appear to be a hamstrung Kubernetes
- I presume you'll be able to use Kubernetes-native tooling
- Docker EE's features are compelling; you keep those, of course, and they apply to Kubernetes now too (e.g. RBAC, Trusted Registry)
- If you're rolling-your-own Kubernetes and are looking for an Enterprise distribution, Docker EE is now an option for you

# Thank you!

- I'll send out a link to these slides on Twitter:

https://twitter.com/lukeb0nd

- ControlPlane are a London-based security and continuous-delivery focused Kubernetes consultancy

https://twitter.com/controlplanio