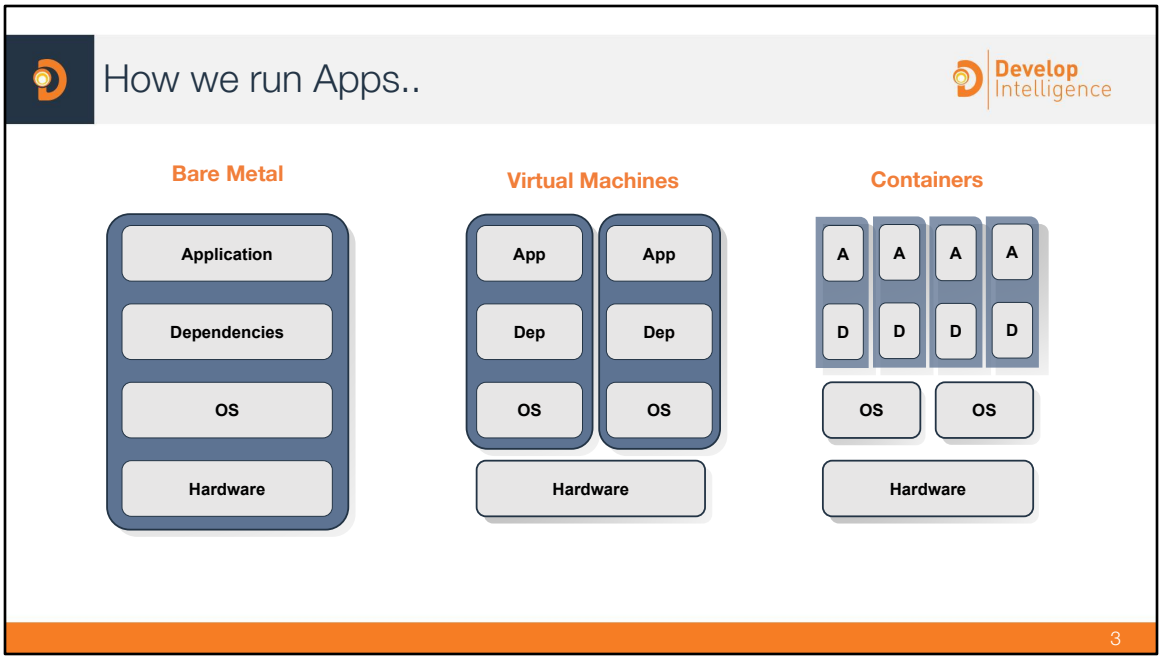
Intro to Containers & Docker

Develop Intelligence

Module Outline

Develop Intelligence

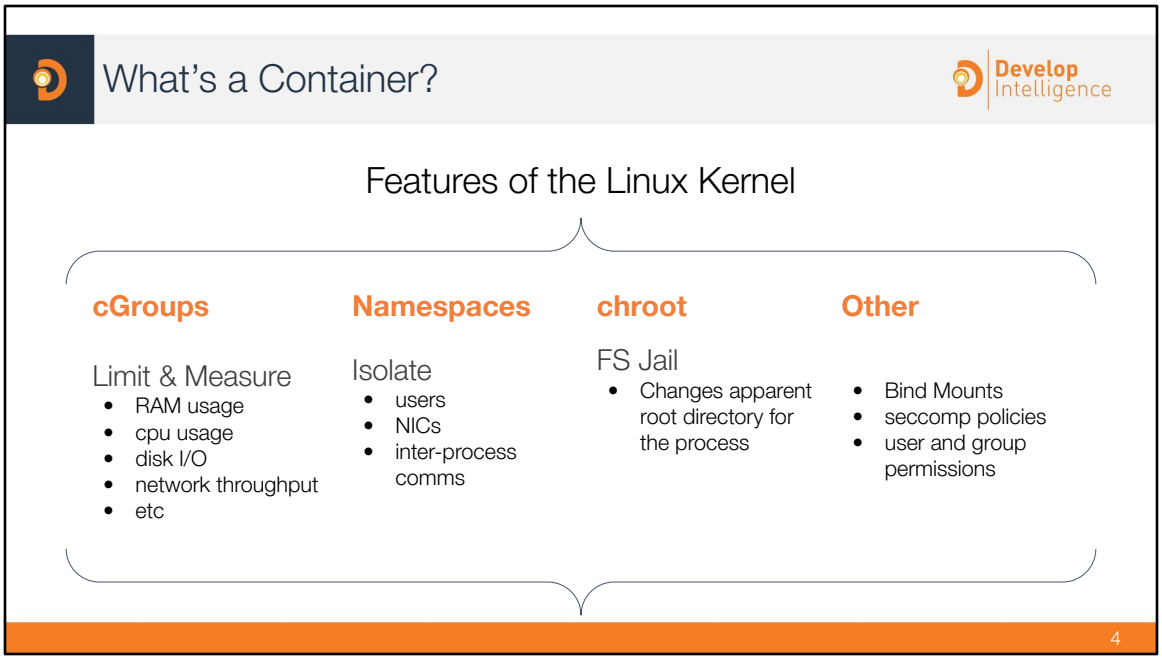
- **Containers**
- Runtimes and Docker
- Why?
- Demo: Running a container
- Lab: Using Docker
- Review



Containers are just a process that's isolated from other processes on the same machine.

Bare Metal = Whole machine
VMs = OS and up
Containers = App and Dependencies

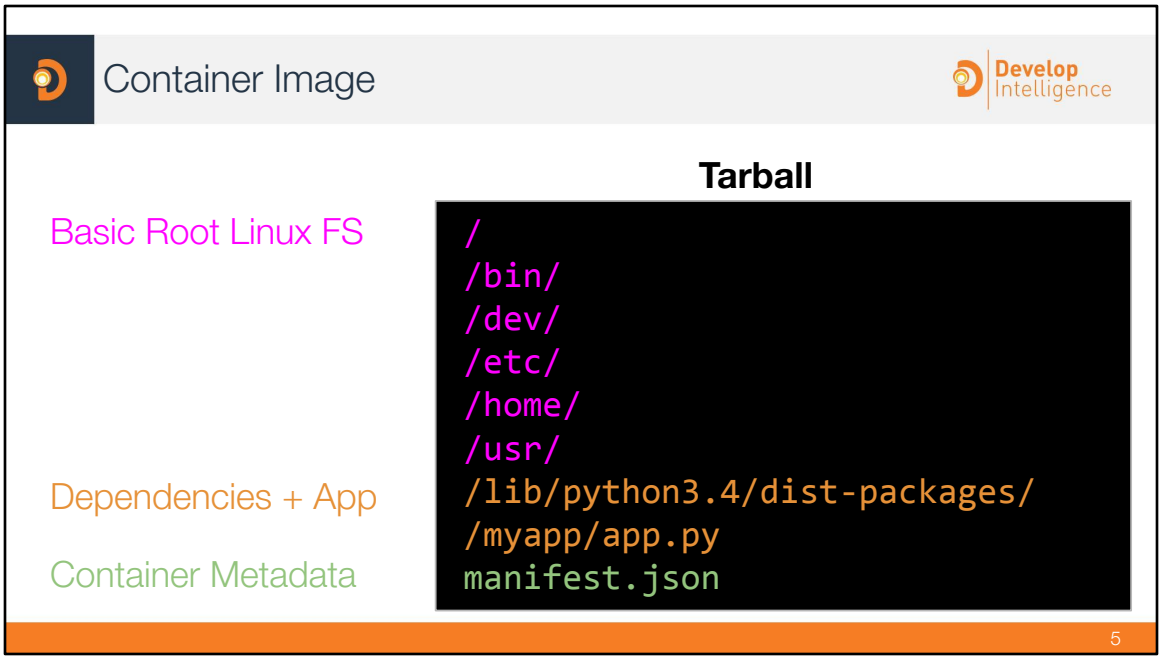
Back in the day, you had to put in a PO to get a server, which may take months, then someone had to rack it, install an os, all dependencies, and your app.
Then VMware and others made VMs popular, allowing folks to get whole "machines" faster and potentially smaller on existing hardware, make this process take weeks or minutes.
Containers get even more density since they don't have redundant OSes, and it only takes seconds to run a container. There is also no virtualization layer between the process and the OS, so no performance impact.



There is no actual "container" construct on the machine. It's really just a process that has been isolated from other processes with these linux kernel features. Since an app is bundled with all of its dependencies, including system libraries, it can run on any Linux machine. It doesn't matter if the host is RHEL, Debian, Suse, etc.

Check out https://ericchiang.github.io/post/containers-from-scratch/ to see how to create a container yourself 'by hand'

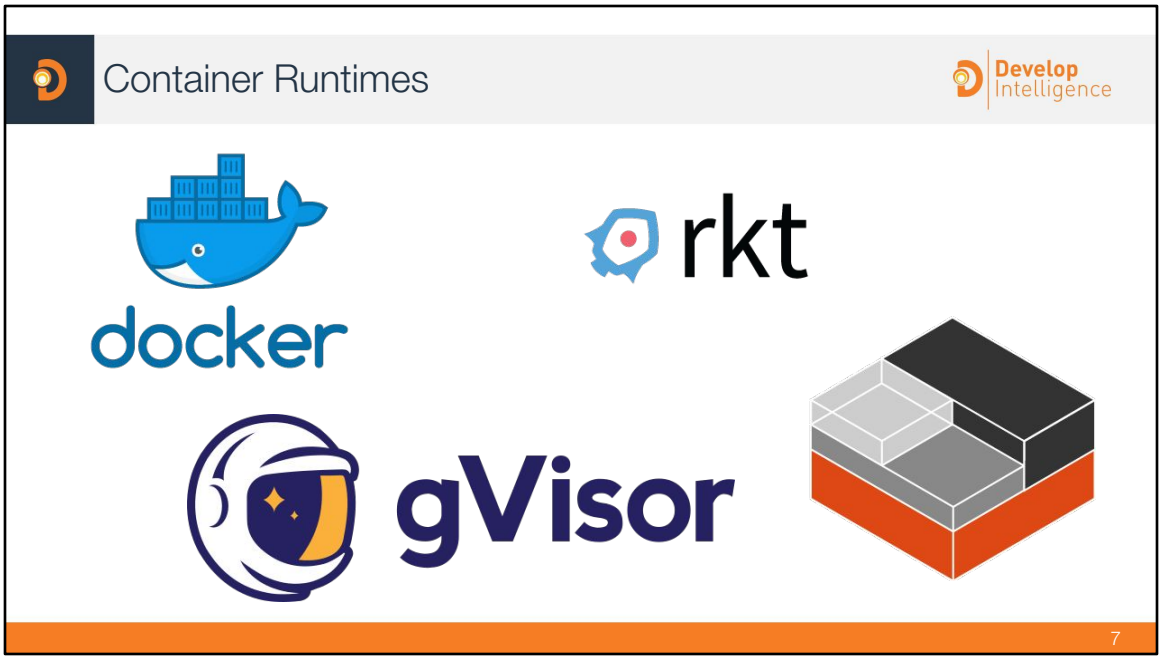
https://en.wikipedia.org/wiki/Linux_namespaces
https://en.wikipedia.org/wiki/Cgroups



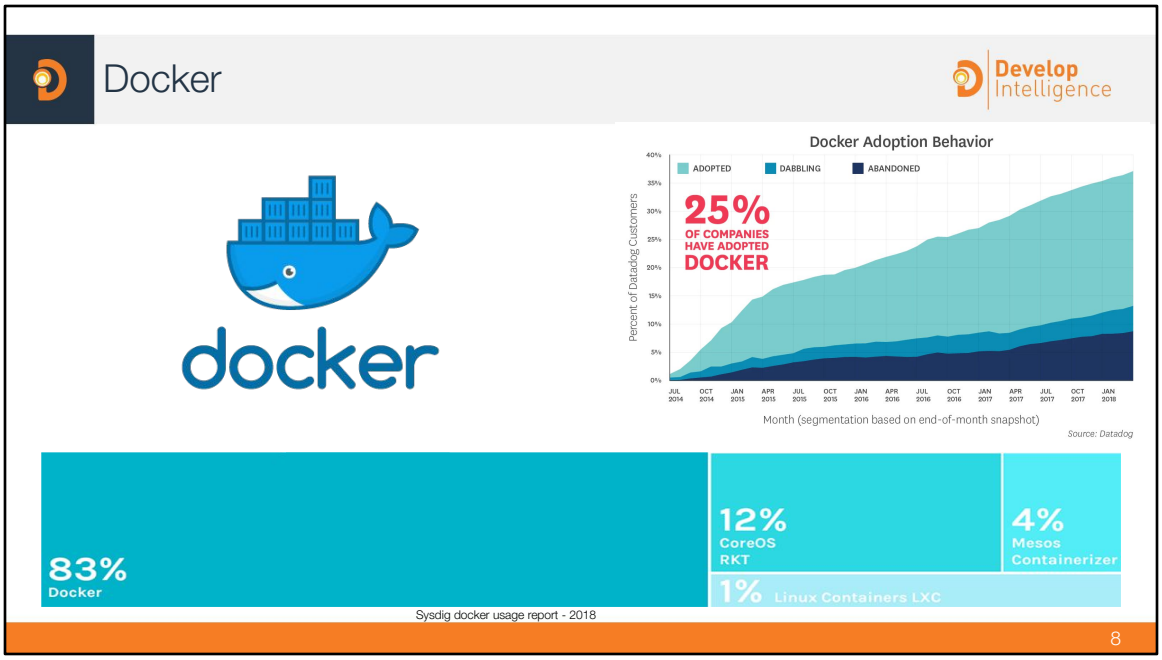
A container image is how we bundle our application and it's dependencies, as well as specify how to run our application in metadata. Dependencies may include application runtime, system libraries, language libraries, etc. This is the filesystem that will be visible to the running application.

Module Outline

- Containers
- **Runtimes and Docker**
- Why?
- Demo: Running a container
- Lab: Using Docker
- Review

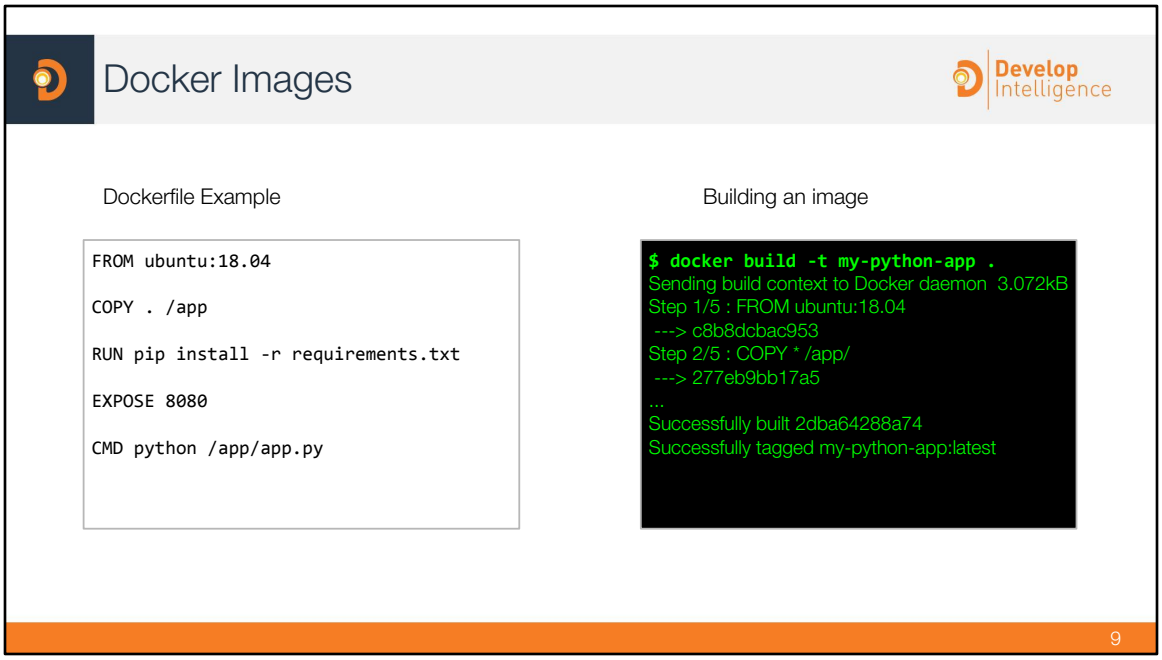


Docker
gVisor
rkt
LXC
...



Docker is a company that created a tool, also called docker, that makes it easy to package container images and run them. They are currently the most popular container runtime, others are slowly gaining traction, however.

https://sysdig.com/blog/2018-docker-usage-report/



A container filesystem is just a tarball.

The FROM is the starting point for our image. It contains packages and files that come with the Ubuntu linux distro (in this case). We do not need to start from a linux distro, this is a convenience to have things like text editors, python, etc. It also means our image is going to be larger than necessary by containing files we do not need for our app to run, this can also have security implications.

Great talk around minimal container images:
https://www.youtube.com/watch?v=gMpldbcMHuI

Docker run command

Develop Intelligence

1. Extracts the image
2. Creates linux namespace, chroots, etc
3. Runs the process

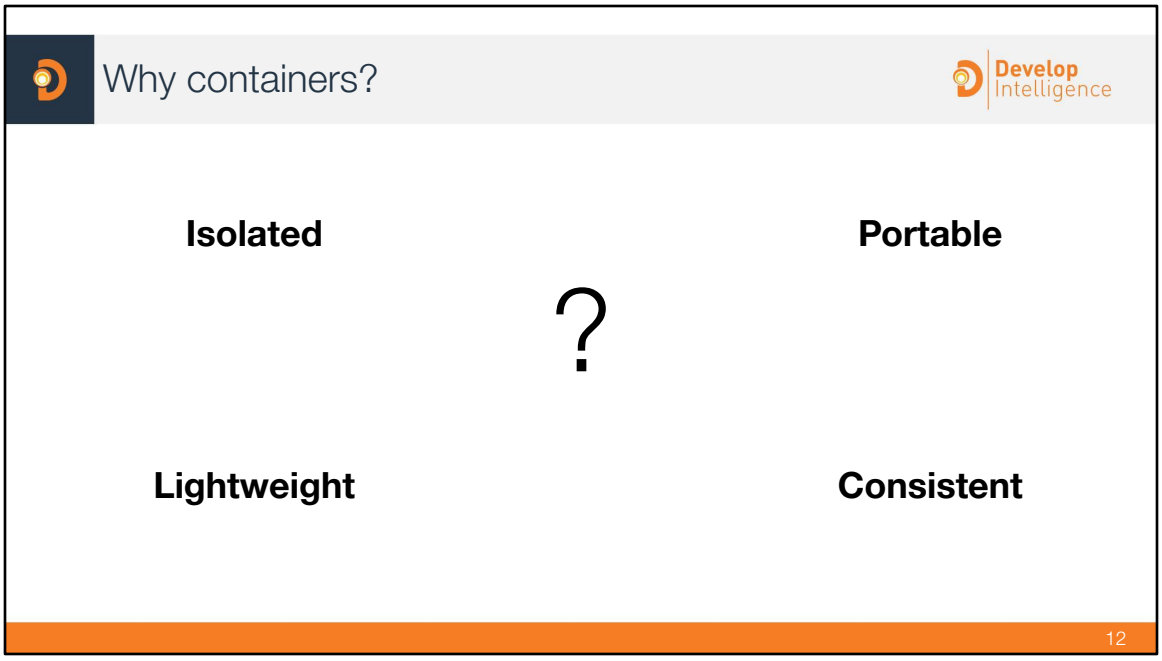
```
$ docker run my-python-app

$ docker ps
CONTAINER ID    IMAGE           COMMAND           CREATED      STATUS         PORTS          NAMES
2c4d58864a71    my-python-app   "python app.py"   2 hours ago  Up 2 hours     8081->80/tcp   eager_tesla
```

Module Outline

Develop Intelligence

- Containers
- Runtimes and Docker
- **Why?**
- Demo: Running a container
- Lab: Using Docker
- Review

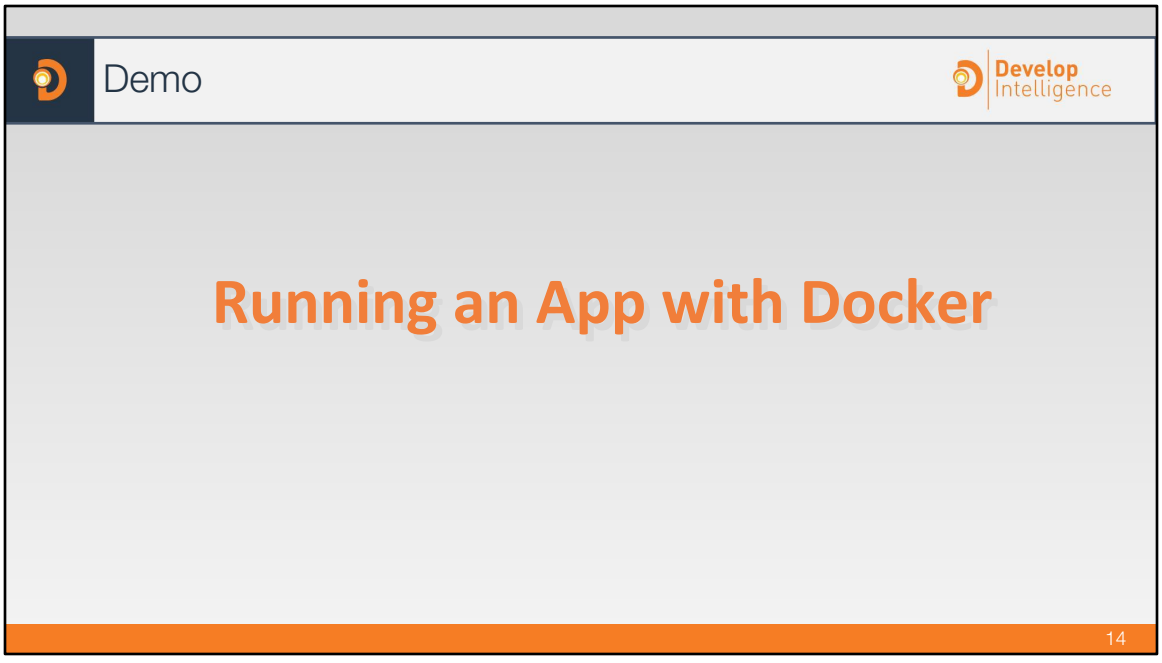


Isolating apps is nothing new, solaris had zones and folks have been building jails in linux for a long time. This was cumbersome and mechanisms to ensure a process had access to all its dependencies were not yet popular.

Isolated - no noisy neighbor, secure
Lightweight - fast startup time, minimal overhead
Portable - sharable, deploy the same way everywhere
Consistent - all dependencies included

Module Outline

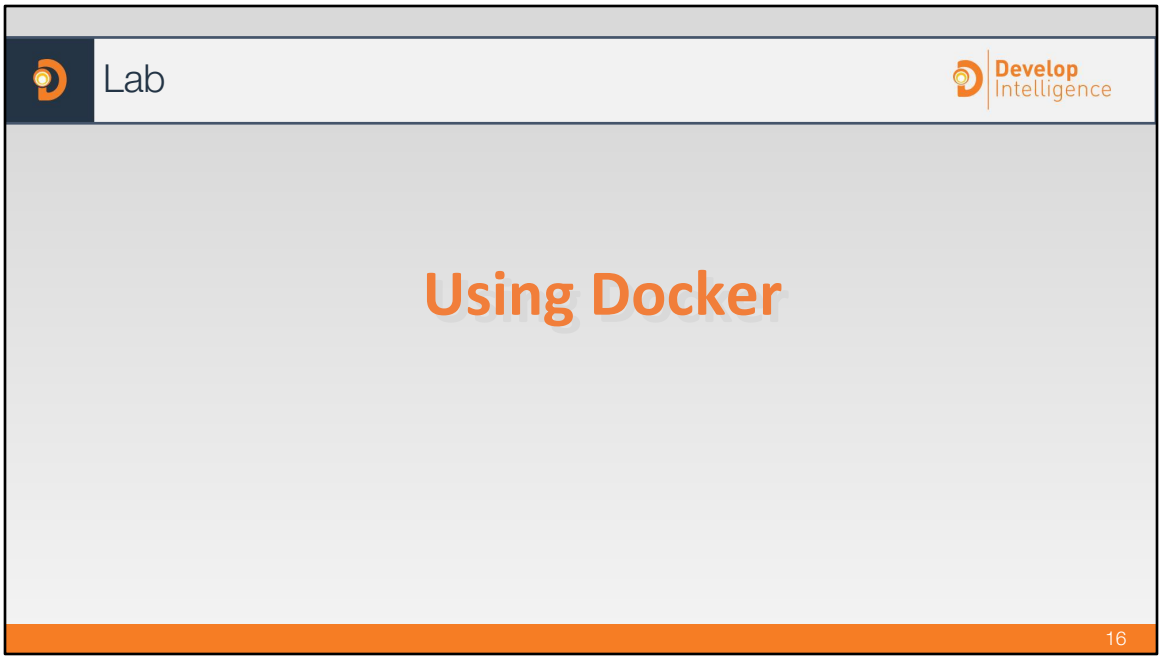
- Containers
- Runtimes and Docker
- Why?
- **Demo: Running a container**
- Lab: Using Docker
- Review


Demo
Develop Intelligence
Running an App with Docker
14

Module Outline

Develop Intelligence

- Containers
- Runtimes and Docker
- Why?
- Demo: Running a container
- **Lab: Using Docker**
- Review


Lab
Develop
Intelligence
Using Docker
16

Note on Windows Containers

Develop Intelligence

- Different isolation mechanism (there is no Linux kernel)
- IIS and .NET applications
- Option of additional Hyper-V isolation
- Docker support, nascent Kubernetes support

Module Outline

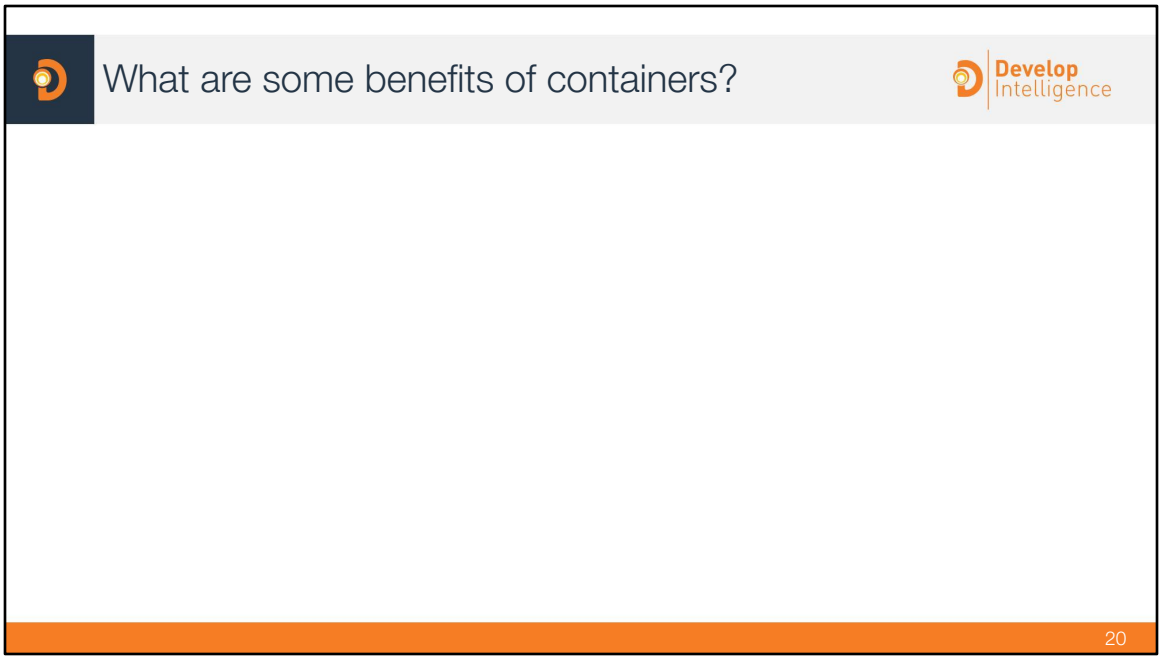
Develop Intelligence

- Containers
- Runtimes and Docker
- Why?
- Demo: Running a container
- Lab: Using Docker
- **Review**

Which exist inside a container image?

Develop Intelligence

1. Application Binaries
2. Application Dependencies
3. Operating System
4. Linux Kernel


What are some benefits of containers?
Develop Intelligence
20

Additional Resources

Develop Intelligence

- Containers from scratch - https://ericchiang.github.io/post/containers-from-scratch/
- Docker - https://docker.io