

Docker & Docker Compose – Transcription of Chat for video telecon, June 25, 2020

15:06:42 From Bret Fisher : <http://www.polljunkie.com/poll/nzcxqg/what-container-things-should-we-talk-about>

15:06:45 From Roger Broseus : Good source for beginners

15:08:10 From stevie b : linking containers

15:08:27 From stevie b : understanding containers once they have been started

15:08:51 From stevie b : remembering the config you have used to do it again

15:11:06 From John Place : other: Multi layer builds

15:11:26 From Mark : Bretfisher.com

15:12:27 From Bret Fisher : <https://www.bretfisher.com/courses/>

15:14:46 From Edwin Wilder : Could you please email the links to the courses

15:15:26 From Brandon Mitchell : Howdy!

15:15:51 From Tristan Thomas : soory i will do that

15:16:13 From John Alexander : BSD Jails

15:29:41 From Bret Fisher : <https://slides.kubernetesmastery.com/#76>

15:32:36 From Paul Ebrey : Lost Audio..?

15:33:11 From Paul Ebrey : back again

15:38:03 From Tristan Thomas : i cant see the slides can make the page bigger please

15:44:00 From Mark : So each container duplicates the OS "directories"?

15:45:32 From John Alexander : You only need to duplicate what your application requires you do not need to copy everything although it may just be easier to do so

15:46:01 From Brandon Mitchell : @Mark all the files inside the container are in something like a chroot jail. There's a lot going on with the overlay filesystem.

15:48:06 From Gustavo : So, the slimmer the OS kernel more layers will have to be copied when pulling a container. I'm just thinking about this because I read many companies are pushing for "thin OSs" like CoreOS

15:48:38 From Chris Ellis : Containers don't contain a kernel within the image

15:49:02 From Chris Ellis : Keeping layers small is mainly about making them quick to pull

15:49:44 From Brandon Mitchell : @gustavo You only save/transfer the filesystem layers once. So if you install a bunch of tools, each slightly different, in lots of images, on a really thin base image, you may use more disk space than using a base image that has all the tools loaded once.

15:49:55 From Chris Ellis : Lots of base container images from distro will keep the base layer as light as possible

15:50:18 From Chris Ellis : CoreOS is also EOL now

15:50:46 From Chris Ellis : The layer caching also negates alot of the concerns over size of images

15:54:04 From Chris Ellis : The container host Oses are more Kubernetes focused thses days. If your spinning up a K8S cluster you're probably using a dedicated OS

15:54:45 From Gustavo : Thanks for explanation. So thin OS kernels are not something that is getting too much traction because not much benefit saving a few megabytes of RAM with the current hardware power/prices

15:56:19 From Gustavo : (I was referring to the kernel OS in the host)

15:56:27 From John Alexander : With hosts with 100's of GB of RAM it's of reduced benefit

15:57:00 From Chris Ellis : There is only 1 kernel, so its moot

15:58:20 From John Alexander : Exactly

15:59:31 From Chris Ellis : IIRC a drawback of Alpine is it use uLibC, which can cause a bunch of issues

16:01:36 From John Alexander : Quite different from LibC

16:01:47 From Chris Ellis : If all the containers you were running all used the same base layer, that would be shared. So the container image sizes are slightly misleading at times.

16:03:37 From John Alexander : Would be good to detail both the grand total and the size after the cache supplies what ever common objects

16:05:23 From Bret Fisher : <https://github.com/wagoodman/dive>

16:05:57 From Chris Ellis : That is a never neat tool

16:05:58 From Dave Compton : Nice. That's my takeaway right there.

16:06:00 From Chris Ellis : very*

16:06:36 From John Alexander : Live that tool

16:07:20 From John Alexander : Like it even

16:08:56 From stevie b : me tool. very useful

16:13:18 From Chris Ellis : @bret is there any support for doing container cross builds? So building an ARM image on an x86 host?

16:13:43 From Brandon Mitchell : @Chris buildkit / buildx

16:13:52 From Krunal : So, when I run python:latest it will check the manifest and get the image compatible with that host, right?

16:14:01 From Brandon Mitchell : You'll need qemu's binfmt-misc setup.

16:14:13 From Chris Ellis : interesting

16:14:15 From John Kennedy : @Krunal - Yep.

16:18:21 From Brandon Mitchell : @Chris if you want to know more, I've got a presentation up at <https://sudo-bmitch.github.io/presentations/docker-build/presentation.html#cross-platform>

16:18:32 From Bret Fisher : <https://www.bretfisher.com/courses/>

16:18:44 From Chris Ellis : Thanks Brandon, I'll take a look

16:19:31 From Roger Broseus : Previous Docker presentations at previous NoVaLUG mtgs. are linked our Presentations page, novalug.org/presentations.html

16:22:09 From Chris Ellis : Nice advantage of podman is that it doesn't have a client server arch. And also support non-root containers.

16:23:28 From Brandon Mitchell : Rootless support in docker is coming out of experimental.

16:24:00 From Roger Broseus to John Kennedy(Privately) : John: is there a way to :record" and save the chat - it's got good info in it.

16:24:06 From Chris Ellis : As far as i'm aware podman doesn't need systemd. it can generate systemd unit files. IIRC podman / libpod uses CRI-O to exec the container.

16:24:29 From stevie b : can you touch on the running docker with cmd line options and then working out what those options were if you didn't note them down. is there a neat trick to work them out

16:24:44 From Bret Fisher : <http://bret.show/securityfirst>

16:24:57 From n mach : Many thanks!

16:25:48 From Bret Fisher : <https://get.docker.com/rootless>

16:27:16 From Brandon Mitchell : Best practice, use a docker-compose.yml instead of running your containers by hand.

16:27:47 From Chris Ellis : You can do networking via slirp4netns with rootless podman, which basically does the network handling in userspace

16:31:03 From Chris Ellis : Excellent talk, thanks very much Bret

16:31:09 From Mark : Thank You Bret!! Fantastic!

16:31:17 From stevie b : thanks. it was when to are experimenting before hardening up and then getting distracted :-)

16:31:20 From Gustavo : Thanks for the great presentation. Learned a lot!

16:31:21 From Ian Crane : Yes thank you very much for your time, kind regards, Ian