Arches Kubernetes Workshop, Second Session

--which is I have an empty minikube cluster.

So is this just locally?

Yeah, I'm just waiting to see what happens with GCP here, because it seems to be being a little weird.

So is there anything I should do to configure-- keep control locally? Or do I have to wait until the GCP is spun up?

No, you can you can just use the [INAUDIBLE] from the one. So if you do kubectl get pods, when you did minikube start, it should have already configured kubectl for you. So if you do kubectl get pods, in theory, if that's all come up correctly, you should have no resources found in default namespace.

[LAUGHS] Yeah, I do. It doesn't seem like a successful response. But yeah, that's what I see.

Yeah. Basically it's a-- yeah. Nice to see you're not doing anything. Hope you're OK.

Yes, I'm OK not doing anything. Yeah, so that's that. I mean, what you can actually do is get all-- there's not really much to see, to be absolutely honest.

And then what is this concept of a pod?

So pod is-- normally it's one container. Sometimes you've got a couple of containers working together. But it's effectively a unit of work.

So a process or a process with some sort of helper process alongside it. But two things that are basically constitute a task or a service or whatever.

And so, for example, if I-- oh, actually, exactly what I can do, so Kubernetes has got a controlled namespace called kube-system. So that's what the dash-n does. And it looks at everything within that namespace.

So within that, there's a number of things already running, which are-- [INAUDIBLE] which are all doing individual tasks or effectively running processes. You'll see most of them have this 1/1, which means that they are pods which can one container. So in other words, it's basically the equivalent of container.

You'll see a couple of exceptions. Here we've got 3/3, so this is actually three containers running together. And potentially you might find that when one of them is exporting logs or doing backup processes or acting as a proxy for some connection or something, you might have another container running alongside. But most--

So it's groups of containers, but oftentimes it's just one container.

Exactly, exactly.

Performing specific tasks or services.

Yeah. And what you'll see when I fire this [INAUDIBLE] up is-- I think most of it I've got is just one container just normally, but [INAUDIBLE] specific reasons that you might want another one.

So that's the pods. The services are effectively kind of a simple service discovery approach, where service has a name and it's got some sort of matching rule [INAUDIBLE]. So from anywhere in the cluster, I can say, I want kube-dns, and it will go off and go, oh, there is a pod which matches certain criteria called kube-dns, blah, blah, blah, blah, blah. And it will reach there. If it finds several of them-- So for example, if I scale it up to try and basically [INAUDIBLE] dozen different containers to look up between-- it's bright enough to just do round robin load balancing. So if I've got half a dozen DNS containers that match the criteria and I try and do a request to keep DNS service, it will basically pass it to one of those six processes. And I don't need to know their exact addresses or anything like that. I can just hit Execute DNS endpoint.

So this is [INAUDIBLE] to one of them or to all of them?

It will pass each request to one of them. But that [INAUDIBLE] cycle, basically.

OK. And you just don't care which one necessarily.

Yeah, exactly. It just a very basic imbalancer really. So the others are probably less-- well, the other one that's worth bearing in mind is deployment. So a deployment is essentially a template for calls. And you can say how many replicas of one you want. So say, for example, I know it's going to be a [INAUDIBLE] DNS load. I could say I want six DNS [INAUDIBLE]. So I create a deployment that's got the template for what that pulse definition, what doc container it's running and what security policies or whatever it's got. And then it says how many of them I want.

And do you manually configure that?

Yeah. So--

OK.

Yeah. So this all gets configured really as infrastructure as code in most cases. So that's the stuff that will get set in the Helm files in YAML markup. And then as it's all running, you can deploy changes to it. And Kubernetes is smart enough to kind of roll this out gradually and not shrink everything while they're coming a lot in theory. So if I-- [INAUDIBLE].

[INAUDIBLE] audience.

You know there's something really obvious we're missing here.

I'm not sure what.

[INAUDIBLE] to the IP locks. Let me just quickly check here because I just saw out there and [INAUDIBLE].

[INAUDIBLE] really is entirely [INAUDIBLE].

Extremely influential.

OK. I think [INAUDIBLE]. So yeah, I can switch back to minicube like so. Be back and look at there. OK. So I guess one thing we talked through the last time is the doctor posed. I just kind of-- I haven't spent a lot of time updating our [INAUDIBLE] to what you guys are working on which [INAUDIBLE] get to. But for the sake of keeping our show consistent, I've just [INAUDIBLE] the same way so you can convince yourself [INAUDIBLE].

So this brings about something that we were wondering-- are you using the same like Docker compose that's in the root of arches or do you have your own custom dock compose file for this?

I'm trying to use the same one as much as possible. You can actually [INAUDIBLE]. So I have [INAUDIBLE]. There we go. That's what we're going to pull from. Give me 1 second here. Then you can [INAUDIBLE].

I mean, I guess if you're using that dock compose, you're probably using the dock file and entry point files that are in there.

Yeah. So by and large, there were a couple of tweaks that help to make, I think. And Let me just see a couple here [INAUDIBLE].

OK. So that's-- OK. Just seize the property of that there. But the difference to 5.4.3 [INAUDIBLE].

Yeah, I think there were a couple of minor changes to be on. And I needed to factor 5.1 [INAUDIBLE]. [INAUDIBLE] of the typekit and stuff I needed to [INAUDIBLE] OK. But again, I think those are more things that could be tidied up so [INAUDIBLE].
[INAUDIBLE] 443 [INAUDIBLE] and not flashing. [INAUDIBLE] cryptography library for some reason [INAUDIBLE]. But yeah, you can see it's pretty minor changes really and mostly stuff that just is artifacts of me trying to do this [INAUDIBLE] and things we [INAUDIBLE]. Now if I go to--

Do you find that the current dock file entry point and dock [INAUDIBLE] mostly meet your needs? I mean, so one thing for instance where I was using it, it's oftentimes-- and you might run into this too. You're probably deploying existing projects, right? You're not necessarily always creating a new artist project. And I know that dock are, I think, entry point file and kind of assumes that you're creating a new project a lot of the time, unless the database exists.
Yeah, I get you. Well, apparently, I was most [INAUDIBLE] first but because we're relatively recent, I suppose, in the grand scheme. Well, actually, yes. So I should--
So I should say the actual what we're using for doing an individual project, we create a little bit more. So I don't know if that maybe mitigate some of it, but possibly so. I can push this up here. Let's see if it's right for me to do this.
And actually, tell you what-- let me--
I mean, I guess what I'm really getting at is that we've been modifying the dock files internally and we have different branches that they're on. And we're wondering would it totally hose your workflow if we were to change this or do you have-- it sounds like you might have your own separate set of dock composer dock files for your work, for your project.
I would say we're tried to keep as in line as possible with the upstream ones. It doesn't mean that we can't adapt. What you'll see here is that we've got the [INAUDIBLE] largest container but really what we've done is introduce some variables or--
It looks like with Docker group artist project latest, the image you have there-- is that just like a standard arches image that you guys created. Well--
Well, this is it. So basically, so we've got the build contexts from the Docker file. That's what we're building and basically the other than those minor like one or [INAUDIBLE] tweaks, the Docker file that we are [INAUDIBLE].
So essentially, this here is pretty much the standard Docker image. I just got that locally from dock arches folder off that branch. [INAUDIBLE] link in the channel.
Yeah.
So that's pretty much a standard arches. Then we do a couple of tweaks there [INAUDIBLE]. That's it. I think probably the most significant thing is [INAUDIBLE].
And it looks like you're copying an arches project from your local if you go back to that. Sorry.
Oh, sorry. Yeah, yeah. Yeah.
[INAUDIBLE] it's like five you're copying [INAUDIBLE] product. So not like an existing project.
Well, to be fair, I did just go through the create project workflow.
OK.
So that's-- it's an existing project in that it was created just before I did this. But yeah.
If you're deploying a project that party exists, for instance.

How would that differ? I guess, and I think [INAUDIBLE] would need to be imported into the DB or elastic. But I mean, the idea of this was actually when we were doing the Antarctic a piece, we wanted to basically customize some of the assets and things which are fairly basic as a thing of the project here. We find that this works will work fine for that. So if I make changes to the actual project-- again, this unnecessary bandwidth. So there's nothing wrong or missing anything. We had to make some changes to the settings local and settings plan file. But other than that, we've dock show that we've got there, just takes whatever modifications you've got in the project and wanted them to create a combined Docker image.
So in theory, I can't see any reason it wouldn't work with a pre-existing project if you modify the settings files appropriately. I guess everything that we've done is kind of assumed arch is 5. But that's the only other thing.
[INAUDIBLE] I mean--
That's been around for a while now. So--
Yeah, I mean, I don't-- yeah. [INAUDIBLE].
Yeah. So I don't really think-- I mean, if I fight off the X of this, this dock with the [INAUDIBLE] this dock [INAUDIBLE] I've got a worker. So the only other things I've done is I've had a worker and rather than queue and [INAUDIBLE], I think as well--
A salary worker, that is.
Right.
[INAUDIBLE]

It was there. [INAUDIBLE].
Had a container called arches worker, I think.
Yeah, yeah. I was just wondering [INAUDIBLE] I had a salary [INAUDIBLE]. Oh, it's [INAUDIBLE]. That's right. Sorry. Of course. So--
You can mount the entry point.
Yes. Does. So the point-- let's see.
You have a lot-- a worker has access to a lot of stuff in a play.
Yeah. I mean, so at the moment, what I've done with the docket [INAUDIBLE] reasonably, standard, in particular, its goals. Well, yeah, this is difficult access to everything. It's not really the main [INAUDIBLE]. The other thing about this is that while the entry point is updated to add-- so this is the first kind of Kubernetes-y thing and it's maybe worse highlighting is as well as doing the dynamic build. I do a static build [INAUDIBLE] Engine

So rather than trying to serve the front end is put a Docker here. So one of the challenges, yeah-- so I how. This Dockerfile dot static, which uses a dummy django settings secret key, but otherwise takes very sensible variables and then runs that in at yarn components.
We've got a settings static that basically tries to make sure that it's writing to standard error instead of a log file. And then that does collect static real. No [INAUDIBLE] collect static real, I had to make a couple of tweaks to get it to work. in Docker builds but nothing dramatic.

And then the last bit is, it copies everything across into a folder. Basically just nginx. So that nginx unpriviledged with those static files and that's it. Which is kind of set up. That then means that, effectively, the arch installation comes to Docker images.

The static in the app.

Exactly, yeah. No for the doc composer I think I just run in, I just run [INAUDIBLE]. But I could have used [INAUDIBLE]. That's what led over to the static. Nginx, obviously, that's not then going update when you [INAUDIBLE] into things. But having got those two, that-- so how,

And then, you're still containerizing? In this particular step, you're still containerizing the db [INAUDIBLE] and all that stuff. I might have [INAUDIBLE] I suppose I just wasn't paying attention to because we're looking at other stuff.

Yeah, so I mean it doesn't really matter too much. Doc compose the only purpose of it really is to try and let me debug stuff consistently with what's happening with your-- but yeah. There's no reason it has to be. So if I go in here. So this is the actual Helm chart. Now this matches up with-- [INAUDIBLE] With this here. Which is, it's still obviously kind of [INAUDIBLE] but this has got the definitions and this values file is one that really contains a kind of default set of environment variables. So in this case, yeah, we set up postgres in the cluster, just, again, they'll be variables.

So does this supersede the environment variables in the proposed file for instance, or?

Well, this is once it goes onto Kubernetes doc, both file doesn't really exist.

OK, for me.

[INAUDIBLE] composed. Yeah.

Sorry say that again, sorry I interrupted you.

So Kubernetes is effectively an alternative to doc compose, think of it that way.

And then does it use, does it use any of the doc files? Like for instance, does it use the Dockerfile?

Yes.

It's communicating. So, OK.

Yes, so all the Dockerfiles it does use. I mean, you can certainly see a lot of parallels between how it's configured but.

For sure.

But yeah, some doc composed doesn't exist in Kubernetes world.

Now if I were to try to get, like we have a dev set up in Docker for arches. If I were to try to get that onto Kubernetes, and I'm not sure why I would, because it sounds like Kubernetes is great for production, [LAUGHS] but just as practice basically. Where would I start? I mean would I start by making a Helm file?

So what's-- the flow that I would suggest is obviously what we want to do is get your Dockerfiles kind of, basically what we were doing following that as much as possible. But for the moment, we've got-- where am I, what [INAUDIBLE] what we're doing. Oh we're good. Sorry. We bare-- no. There we don't go. There we go. OK.

So we've got this Docker compose, now that's where I set it up, is that I can do Docker composed builds and that will give me those tagged images like the Docker compose. Yeah, there we go. So I can basically say, I want a Docker group, is flaxandteal art project is workshop to and then doc group was built and that will give me those two Docker images. The static one and the nonstatic one.

I'm not sure I follow the commands. You have Docker group and arches project as variables. But before your command, I'm not sure I follow that. Is it?

Well, as in, just sets out what will be used in the doc file. So if I go, all right, if I go in here you'll see we use the--

So maybe it's just, I think it's just the order you have them in the command file, in the command line that's confusing, but yeah it's fine. Whatever.

Yeah.

It's the same command line argument, it's just you put them before the command.

Yeah, yeah.

Yeah, OK. OK. OK, so that brings up those two.

And then basically I can push those over. So in this particular case, for the sake of simplicity, I've just pushed the dropper up. Obviously you can configure private Docker repository or whatever else. There is a way of doing it with mini cubers, just doing it locally but it's not. Well, actually recently [INAUDIBLE] it's not too much.

So first [INAUDIBLE]. Then, that's basically as far as your local stuff is concerned, that you've done and dusted. In continuous integration, that basically runs as one of the steps on site GitLab or GitHub actions or whatever. So from that point, the Kubernetes cluster, wherever it is, is just responsible for going [INAUDIBLE]. And if I go to that record that I'm showing you. So I've got setting up. That's there. Yeah.

So. This, you can clone this now. You can actually do it strictly speaking, you can do repo add arches project and we've got it basically a GitHub. Which is actually going to pick up variations to create , this, so it's kind of almost like a package repository but for this arches project chart. A chart.

Are there different helm files for say a package repository?

Oh sorry. So--

Of helm files? Or?

Yeah, exactly. So you see there that is flaxandteal.github.io/helm arches. So that matches up to basically just saying, get the templates from this. Get repo.

OK.

So, really what you're doing is it's putting up the Docker images to some sort of registry, getting templates from here. And then the third ingredient is whatever secrets you want to override it with, knowing-- or other config that you want to override with that's non-standard. So--

And that's what you put in that helm config file that you showed me?

Yeah basically. So I can-- there's no real way to doing this about simplicity. [LAUGHS] Let's see if this works. It will do better at demonstrating once we actually have running. [LAUGHS] So, in theory.

That's new. Just to see what happened here. But just as exciting results trying to connect to GCP, sometimes connections to GCP. Oh, yeah OK. So helms doc, yeah. OK, that was much faster. So now, I do this kubectl. Now, I haven't configured this so it doesn't know where those images are or anything. So it's just a default setup, I just want to sort of show that effectively you can get a default set up with two commands. So that's how to reboot and then how to install. And that's already starting to create some of this stuff all [INAUDIBLE]. Its image pull back off, why is that?

Oh, OK. So yeah, OK. So basically because I've used just a standard call think without really configuring it, [INAUDIBLE] what you're trying to do. Let me just take a copy of that values, those default values and then fill out one of these here. [HUMS]

OK. So this file is found here. And then that shouldn't be-- repository, right. So I pushed off that workshop too. I might try and reach out to the moment of truth. Static. 5.1. [INAUDIBLE]

You're ready now. OK. And it's-- that's right. So now I can actually, so I've just made some changes there. I can actually do upgrade. And I use by custom values. One million. Well, [INAUDIBLE]. [SIGHS] Oh gosh. [INAUDIBLE]

So some I think I've just changed versions accidentally there. Right. Now, so all right versions. So hopefully that's [INAUDIBLE]. You're doing the same thing, except with those [INAUDIBLE] do you use those new images?

Mhmm.

Now they're ready to open enough memory to try and run Elasticsearch tubes, so it's sitting panting. [INAUDIBLE]. Oh. Oh. I think I've got two versions there.

I may have to cut this short in a few moments. I probably have to go relieve my wife.

OK. That's right. I mean, what I'm going to do is try and there's a couple of scripts I put together for this really, which is going to try and finish up. But I think having the mini [INAUDIBLE], of course. Those are currently downloading those images. But basically what I should have at that point is to please-- [SIGHS]

[SIGHS]

You ready to [INAUDIBLE] today.

[LAUGHS] This demo.

Yeah you know what, you probably not wrong. I know why it's telling me it doesn't exist because it'll literally be connected to. [INAUDIBLE] All right. Well, at least that's [INAUDIBLE] That's still creating those. OK, well. Essentially once that comes up, that should be available.

Yeah, I see the other two to get. It's basically the standard arches. You own that. As you can see there, got couchdb starting, postgres, and rabbits. And then, within that values file, it's kind of a conflict for those so you can override even the bits of the postgresql thing and find it's running and [INAUDIBLE] and so forth. Which means that you can-- which means you can point it to other places as well. And so, if you want to [INAUDIBLE] start yes, for example, or several times.

I could figure that.

Yeah, exactly. Exactly. And also things like, so we use a variety of approaches you can use. For this one approach we use, is sop. Which is basically a way of running encrypt secrets and actually there's kind of a nice format for providing an encrypted secrets file. Again, you can do this in CI. And then not a secret file is [INAUDIBLE] structure. But the secrets that need to get passed through to individual processes are individual [INAUDIBLE] or individual crawl jobs or whatever. And get those from there.

So I've just simply been pulling it. Probably because I'm on 4G here. Just [INAUDIBLE] I mean, I suppose probably what makes sense is if I get that, it was kind of set up scripts. I've spent at the house quite a bit of time to work on those solutions should be useful to help get it set up, at least a Minikube. [INAUDIBLE] to do is if we can bounce back and forward, and get you set up as well. Then we could maybe do a bit more of a dive into the limitations of what needs fixed and see if anyone else wants to join us. I know the [INAUDIBLE] folks have been looking into this in parallel as well. So what we could see if we can get.