Arches Kubernetes Workshop

So OK. That's just to confirm no objections so far. So if you do, you just say. So I guess the key practical points that's what is Kubernetes. And I realize that there are observations that cover this as well as it's a way of deploying infrastructure, the roads have been cloud specific to our standardized. It's open.
It's something that's sustainable as toy vendors are potentially evolving their offerings that there is a better a vendor and way of obtaining this.
And it's also something that's got quite a natural set of progression from taking from something that's working with Dr. Kapos up to something to cloud. And then the evolution can continue from there.
And it does have a lot of structure in how it's operated. And there is a well established process, vendor neutral process for evolving how Kubernetes works. And it's something that we'll probably get into.
And then also we tend to as much as possible, try and get our infrastructure into all of our infrastructure's code so that we can set up a new platform that's straightforward.
So we also, where possible, like to get that drawing from CI added use in care form [INAUDIBLE] as well as if we were able to make infrastructure changes and pulling updates to the infrastructure.
For example the new components, optional dependency is added to Arches. And it's something that we can pull in to get to what we're happy to [INAUDIBLE] place through report option.
It's also something else particularly that we were seeing a lot in digital outcomes, tendering, and things like that in the UK. And they're quite a while.
Key points around Kubernetes definition, everything is in YAML definition for those of you-- imagine from what you said, you at least can have a look at some of the darker cool stuff. But there's a relatively standard easy to read syntax compared to dark one in particular.
And we've been using Kubernetes on a variety of platforms. And one of the key advantages processes transferability. And I think Andy mentioned that going on to Azure. AKS is something that can have its challenges as well, but it's a very standard way of deploying some of the AKS in [INAUDIBLE].
Mentioned CI, but that for us is not the big thing. What we'd like to get to ideally is saying, OK, we want to add a plug-in or change something or in an artist project without pushing it to get us enough to get it live.
And that's something that can happen in staging et cetera without having to restart infrastructure around [INAUDIBLE]. And again, the abstraction from the physical machines.
So we've been experimenting with this for a while. I'd find that setting up a running Django and [INAUDIBLE] find and then set the work.
And we hook in the air collector as well we'll actually know look, how do we add in Nobex and make sure that we've got HTP going and make sure we be able to have worker processes. And that's something that we can then start structures to definitions and [INAUDIBLE].
So the approach sense the Kubernetes uses to allow simplified service discovery is that the individual services are separated out and they can be addressed on [INAUDIBLE] results that can be addressed as in through to however many replicas.
So if you scale our dimension x server or matchboxes, that Kubernetes will handle the route between them. Also REDIS, for example, if we want to scale our REDIS, we can say how many we want and then Kubernetes will manage a lot of the readings with it.

We're not having to really think about an actual process. And I think sometimes it's that one of the big advantages, is if we can separate out the state from stateless then scaling can be extremely straightforward. Also, I'm going to touch on this a bit is doing things like autoscaling the install custom metrics, and things like Django, and so forth. I had mentioned in chapter minikube, I did spend Saturday and Monday trying to brush up some stuff and get it working on minikube. I did get there yesterday, which I will show you. There's still a couple of things to go.

And it's actually quite a good chance for me to show you where I hit a couple of challenges. And I'm curious to know if there's things that I'm doing wrong. Or if there are things that would be good to [INAUDIBLE]. In particular, it's working behind the minikube [INAUDIBLE]. On my host laptop, for example, it runs a virtual machine. On that virtual machine, it runs one Kubernetes node. That is one Kubernetes VM.

But other than that, it's perfectly normal Kubernetes cluster running on a VM on your machine. So it's a great way of testing deployment books and stuff like that. But it's probably the Kubernetes equivalent of Docker code. But as it runs with VM [INAUDIBLE].

Another component that we are very keen on in general is Helm. And Helm is the Terraform of the Kubernetes world in that we can define-- this is taught not just project-like. Some of you may have seen the charts where we've been trying to put together a so-called Helm chart, which can then be pulled in. And you can set your parameters to install Arches project on my Kubernetes cluster. And it just pulls everything in, sets up et cetera.

So we'll come back to what we've got. And [INAUDIBLE] anyway. But I'll give you an example of some code. Each process runs in a pod. Pod, there's normally one, but can be more containers. So just as if you're running them locally. And if it's more than one container, normally, they're extremely closely related. So something like a process of the log exported for that process. And that can run together.

And in that case, we can add some labels to it. Kubernetes is great for being able to manage things with meta data. And so filtering things by labels, pulling logs by labels, and [INAUDIBLE] updating, rolling back deployments, separating things out, giving connections, et cetera and using some of that metadata.

And then the rest is actually stuff that should be relatively familiar from Docker. So you've got an image [INAUDIBLE], and it's certain parts that are accessible outside of the [INAUDIBLE] space container. I also have a-- Let's go there, we also have a service [INAUDIBLE] and as I mentioned, the service gives away that even if we've got multiple quotes, multiple engine instances to a balance between service which is one way that one particular DNS entry that we can target and we can get between.

So here we've NGINX and that will be able to go balance between as many pods instances mentioned. This is particularly important in the context of deployments where we can define what are deployable definition.

I moment should can have multiple instances of next deployment is essentially the grouping of those, and we can say how many [INAUDIBLE], because we can scale it up or down the scale, and it's a little bit like an older scaling equipment if you ask so, and so forth.

And then we define a template that can be deployed, that describes how you want it to look. And then when we update those capabilities are smart enough to be able to rule out a deployment by restarting one at a time without killing them all at once that sort of thing, so that we can check things are alive and healthy before killing the next one on allows us to do zero down time on deployment.

So, I guess [INAUDIBLE] and the way that's deployed it's helm install, deal with [INAUDIBLE], but we can show what we've done manually locally, and we put together a series of those YAML files with template language, and we put together a configuration file that maybe has specific settings for environment, for example, for the [INAUDIBLE] or the Arches project, that sort of thing.
And then that creates all of the Kubernetes entities, like those deployments, and services, and so forth.
On this topic here, on the left side of it we've got essentially secure configuration, and on the other side we've got a helm template of YAML files in this documented YAML files can [INAUDIBLE] get, and so we have such a [INAUDIBLE] kind of about the answer what actually happened to store passwords and so forth, [INAUDIBLE].
OK, continuous integration, we tend to work mostly with GitLab with the hub and that's because of working quite nicely with building Docker image, especially the images and then updating Kubernetes to closing, so that's you get deployed, you can watch like a one three, and also that we can trigger jobs on the [INAUDIBLE] for example those kind of commands not just the [INAUDIBLE].
So few considerations that we kind of picked up on, you this morning and unless we've got to spent some time on it, but really looking at some of the things that the kind of cover and you're going in normal ops where what challenges are that we would see when deploying [INAUDIBLE] normally and how does it translate into that.
How do we-- are the ways that we can actually get a benefit by saying, who listen and there autoscaling will be set up in a sensible that it's how the Arches community recommends without you having to [INAUDIBLE].
OK. So that's the three times faster summary. Just to make sure everyone's seen that fresh. Now, tonight we want to [INAUDIBLE] hear. So I will-- Yeah. Is there any questions to begin with? Or anything on that so far? Or shall we go on to have some practical?
No questions for me.
That all sounded--
It's all good.
Yeah, I'd say I mean obviously, you've got a lot to be familiar. But hopefully-- OK. I can talk through a bit of what we've done with kind of what we find in the arches and we hope you see the Docker images there, but I'm conscious that those people are able to see those, so I don't know Ryan, if there's anything you want to show or comment on? Otherwise I'm happy to talk through here.
I don't think there's anything at the moment, I think our work is a little bit of a divergence from what you were speaking about, because we were focusing sort of on the point arches for development in Docker. So maybe you should go first and I'll see if there's anything I can add from my work.
Scored at the time. OK, that's that sounds good. I mean realistically what we'd like to do is try and make sure that we keep as much of overlap as possible. So I'll show you, if I were trying to do that, because in the ideal world, we want to have a nice smooth flow of lifting the environment, so that as similar as possible that's been one of the options.
OK. To what we have hear, now I should contextualizes this. We've been looking at a POC risk basically, which is very early stages but doing some Antarctic work, and we kind of thought this-- try if we can put this into our flow, and this would be really good. So we've been kind of working on that reiterating that, and I think it took us a little bit away from making sure we get a project set up from scratch.

So we've been trying to do that over the last week and just see how that goes. There's still some rough edges around that, but then that's what happens in a workshop, and I'm going to use that at list I'm going to stick to it. So let me see if I can pull up.
We got here, Anyone's-- if anyone's found it hard to use background radiation I can switch from the point of taking advantage of having my laptop close if it's not destructive. So who's to say?
OK. You can see that right?
Yep.
OK. So--
Sorry, I think you are in the Zoom panel is showing black for us.
Oh, is it?
OK.
[INAUDIBLE]. Sorry to interrupt. There's a black box right across the terminal, and I thought maybe it was the Zoom interface for you, but perhaps not, sorry.
It tells that sometimes, if it's [INAUDIBLE]. OK, as long as you can all see-- yeah, if you can see [INAUDIBLE] that's good. Yes, I am trying to be multilingual with my video conferencing interface so it's technical but practice, unfortunately that's also decided to put the mute.
OK. So what I did and I'll try and get them mixed up here. So I've taken would have been doing is working off the arches at one point O, tagged as kind of where I've taken the code. I've added let me just check. One or two minor changes to try and avoid having some difficulty building with YAML, a couple of those. I think are just going to be made of tidy up and also I'm just giving some conscience, so nothing too major and things that can probably get rid of really.
So essentially, this should be working on what's there, not to be one of our objectives is we want to see if we can build this without having to-- I mean, put on some project, you see. So Docker file here, is what actually rebelled, I'm assuming from what you said that everybody's done some Docker work [INAUDIBLE] with Arches. So that's just pretty standard Docker from Dutch from purchases [INAUDIBLE] that, and we built a local version.
Now, there's a couple of handy tricks that we can use for this, not [INAUDIBLE] the way we do not-- again, this is part of what we wanted to discuss. I started a minikube instance, so that's shown at the VM locally with commander there, and that is giving the-- well for quite a bit of overkill for people giving you a nice big EM [INAUDIBLE] space so that we can run that very light as well, but that gives us something to work with.
So if I make something within context that you control, I'm going to use that as a kind of standard on what we've got here. So here's what I did earlier. This is deployed nice and light, one instance of things and its components and-- Yeah, sorry is question there?
No.
And the two containers that are specific charges that are standard for the HTTP since they're separate Django built and then static, which is I guess this one of Django's is getting assets pulled out in reasonable and a sensible way.
So rather than going down, haven't looked at CVN approach as yet. We thought, well, we'll start with the Democrats actually put them and amendment the next process to begin with, and in that way, it seems to kind of open them to container [INAUDIBLE] them, but we've also done some projects where part of our special instructions to the [INAUDIBLE] process of [INAUDIBLE].

OK. So, to build that we've started with building the Docker container in this [INAUDIBLE] as our case. We then created two Docker files to build off that. So this one here archer's latest is the direct don't stop beginning from reQL and we've gone through series of commands, so one thing we've done is [INAUDIBLE] with privileges.

And we can take the arches project name as a variable into the Docker built, and then we run our YAML install, and copy in our project itself, and the project here that I'm using is one, all I've done is gone through the quickstart steps to create this, to generate this, and I've added a couple of just in a second, copy that in the container, run YAML install.

[INAUDIBLE] and then we set the entry point, to the entry point, this will help with the container, [INAUDIBLE] which is just out. We've also got a static Docker from sticks and keep up with this latest [INAUDIBLE] yes, and this is a work in progress.

So what I've done here, is again we take not just project, we pull from the dynamic container so, when I drop the dynamic container the Django I take a quarter of arches underscore protect, in this case, not just my project, [INAUDIBLE] and switch the use actual route, sets a series of dummy variables just basically collapse from the processes to generate their own components, and collect static files.

Now there's a couple of minor modifications I've made to the entry point by inserting custom one for those who are familiar with that. But essentially what we're doing here is just trying to use that to use one's container to export all the static assets.

And then each time we want to put up activities somewhere we rebuild this container, and then because we don't need for hosting static assets we don't actually need the whole container, then we switch over to the next container and publish container, which generally is the [INAUDIBLE] versions to produce a [INAUDIBLE], And we copy in all of the static files from that previous step basically this becomes [INAUDIBLE] static images.

So it's kind of a simple way of getting our sonic assets separated. To make sure we can run locally, and this is probably one of-- it's good to compare notes with what you're saying Ryan. We've got a modified version of Docker rules from the repo which build that dynamic container I showed.

Like I said, we can pass an argument of the artist project into all of this, so my project in this case is what I'm using. We set some local environment variables for the local depth, this cannot all be overridden once deployed, and then we set up a new relationship once really just Docker.

To start let's just bring it up locally at list on those machine, said a worker about a one salary compound there just so you can easily run them, worker process for it gets passed on to it, and then I the added salary somewhere down here. [INAUDIBLE]. That's because that works the same way as the arches container for the arches Docker frozen the repo, then it can rebuild ED from the damage.

So it's [INAUDIBLE] but as I said, I think it'll be good to try and put a little moderate improvements or patient development with the government and for [INAUDIBLE]. OK. Now, there is a number of considerations around building production, which is building the opportunities, and so forth on top of that risk, but just keep an eye on time, this one.

So I can bring that up with arches projects, equals in this case my project, slightly further back I thought. So arches project, my project, [INAUDIBLE] and that will build my [INAUDIBLE] container [INAUDIBLE] computers and so forth.

I've also wrapped that into the old script here, just too much is going on, but slightly more complex stuff the line. We've got a dummy secret key there just running the disposable static container that scoops backwards during the build process, and running those tickets as other containers.
And there's a nice little trick we can use to use minikubes Docker registering is a rather a basic remote being put back into the end. Minikube as a handy little command that we can run, and then if we do the build there to build it some of the people have already see it with the help of insurance into that.
So if I do Docker images, I've got quite a lot of images there, but there should be a project. Here we go. So there is a project start container and an arches is my project container. Those are quite big, the static ones right, but what of the my project is pretty big.
There more of them to set up on. [INAUDIBLE] You'll also see quite a lot of other images have to be to hand for the phonetics when it's actually disable connected to the actual around this cluster internal registry of many came so you can see a few of those.
And [INAUDIBLE] so forth are down there as well inside that room. OK, that's actually it for our Docker setup. I know I went through that relatively quickly. Does anyone have any questions on that? Or any feedback, thoughts? I'm going to say that a good [INAUDIBLE], so go ahead Ron.
I guess it's not so in the Docker setup. I was going to ask, where do you store the Kubernetes or minikube configs, the YAML files you spoke of. Maybe you're going to get there now.
So yeah, exactly. So that's that kind of gets us to the point where we can say, OK, we've got two containers ready to pull in, and then yeah. So Everything from here on, we can kind of forget about the arches project code, we just assume that it's all in containers, and so it kind of sort of ideally separates into two components how I use Kubernetes definitions.
So given where we're at with Docker stuff maybe I can hop in now and talk about what we're doing with Docker, because we're definitely, we haven't started to mess with Kubernetes and minikube get.
Yeah, please.
So host has disabled participant screen sharing.
Oh, sorry. Give me a sec. How do I do that? Does that work now?
Here we go. Just to open the window. OK, I should be sharing my Vision Studio code window now. So just to quickly run over. This is going to be-- I mean it's going to look really familiar because it's similar to what you just showed of what you did. Basically our goal with Docker was to set up containers for projects.
So just like you guys, we took the Docker files out of core and modified them to work with arches, and we ended up trying to work with our projects.
We ended up because of context issues having to create like a Docker underscored project folder, and then putting core arches next to our project just to create the container to pull in core arches which needed to live next to the project in the container.
That being said, starting with our Docker Campos here, this is are arches for science, which is a pre-existing project. We add the Docker file in the router-- rather than a compost file and the route of it.
We define the container here, one difference from what you are showing is that we just pulled out our environmental variables into this M file, but as you can see they're all very similar to what you showed and they'll apply to their various containers.

Similar to what you were saying, all of these containers down here are pretty standard containers. We added the suffix of the project, because we oftentimes in development are running multiple projects at once here.
So this was the strategy just for simplicity, knowing that maybe in the future we're going to use like one database container, for instance, for multiple projects. But as I said this is a work in progress, and this is simplicity, we did this.
We also for this particular project added a cantaloupe container, which is a dependency just for this project. So this exists in this document file, but I went to arches here, another project we're in for, it might not, or we something a different dependency for that project. Let me see here. So I think that's most of this stuff we pointed to the cantaloupe volume, and the dependencies.
In the Docker file, this is pretty similar to what's in the repo, except we made some in-- I guess I should say the core arches repo, we made some modifications here to be specific to the project, but not a whole lot. I don't think we added project dependencies.
In the entry point file, one thing that we did is we created a settings Docker file, or we took the same Docker file. And as part of the entry point for file, we copy that over into our projects.
So let me see I don't see it in here. Down here it's here. So now we have a setting stalker file here, and then in our settings file in our project-- oh, what are we doing here? Then in our settings file in our project we import this Docker file similar to what you guys are doing.
And I think our entry point file handles running the dev server and stuff like that. So I think that's mainly what we did to customize it for each project. As I say we focused on just development.
So we don't have noticed in the buzz file like NGINX or Apache because we're just running the Django dead server for now, although production will probably be the next step here once we clean up some of this stuff, our entry point stuff.
So. Yeah, I guess from here it'd be interesting to as I said adapt this for production and then integrate it with what you're about to talk about with Kubernetes.
Cool. Thanks Ryan. Yeah, I think it's really interesting, particularly, I think some of the things that we've come across probably quite similar and we've had a discussion once to back around some of those considerations too.
It's really helpful the setting is actually an interesting one as well you would find some of the pitfalls that we came across were just trying to get the settings files as much upright.
And little things like just getting to the making sure that the logging is efficient console instead of local storage, and that sort of thing. In that context. So it might be worth going a bit deeper at some point on some of the specific changes and things.
There were a couple of things where at least the way I was approaching it I started hitting circular dependencies trying to get darker and arches to and I was thinking maybe it might be something where you guys to be able to say this is the way it's set up definitely but we've got to follow. So that'd be good.
OK. Cool. Thanks Ryan. Does anyone have any questions for Ryan or any questions out there. No? OK. So I'll go on to look at some of the Kubernetes side.
See 20, 40, 50, time. Perhaps you just what time all right. Let me see. OK I'll back up again. I've got Ron opposite me here, so he's able to nod every time I ask a question. So instead waiting for [INAUDIBLE].

So, OK. That's arches file folder like I say about a couple of things that are similar to what I saying, we'll make a couple of what is the entry point, because we're sort trying to go down that sort of full setup. We've got-- we've done a little bit more on salary end. And so we got a wrong [INAUDIBLE] to that.
And so that gets out of line think that's actually the only one [INAUDIBLE] out of the budget report. You see there on my screen, I mentioned adding into the local settings, just to force the logging, and if I open up local settings a few just bits and pieces here. So one of the things was having to select improve this by making the allied hosts.
Broader because we're getting IP coming in for the health checks and things like that, but we have quite worked out how to handle that, but obviously, that needs improved, and then just pulling some of the bits and pieces from the environment. So that's actually really it.
But on the other side of this, we've got Arches helm definition. So we now assume that there are two different containers. Well, two different containers, one with Django and our project built into it, the other with NGINX or static assets built into cold arches, my project SQL [INAUDIBLE].
Within the home chat here, we have a values file that gives kind of the option of values, and we've started from kind of a classic skeleton, and added I didn't think relevant to arches. So for example, we got a domain names so that we can add those easily in those get dropped into the right places in the config. Helm will let us template out our config files in general and then they'll get pointed in. So just because something that appears in the settings or so forth those are things that potentially we could have as a template that could get much into the room on time if that is desirable. Then better on PostGIS than what standard of HTD evidently looking wildly exciting.
To get this up and running though and making sure it works on minikube, you just need to share it. That's [INAUDIBLE]. We have created this chart YAML file with a chart level cloud, and that sort of gives a bit of a template for them.
Again, this is like 0.1 dash alpha that is what I call this to make sure it's unambiguous, but it's not supposed to be. So elastic Postcodes coach [INAUDIBLE] then points out that [INAUDIBLE] there is one to [INAUDIBLE] project. And then we can say helm.
[INAUDIBLE] up there looking to just in case of a control in the live demo. Call me a power guy, but this is a workshop. So I'm going to avoid breaking everything just what I'm trying to demo with.
Do a rerun afterwards if you want to see whether resilience worker is, but we can update tendencies that are pulling down from some positives not things like the equivalent chart for postgres and separate chart for college.
So those can be purged and integrated separately by their upstream teams. That's great as well of course, because it means that if they are recommend deployment structure changes, which it does from time to time, for example REDIS might change from recommending deployment as a cluster of one master to replicas, to [INAUDIBLE] or so forth.
You may also want to pin it, while it's some traction with one bit, but it kind of gives you a way of saying, OK, what's actually recommended deployment architecture for each of these dependencies? All right so that pulls us in.
And then we need to, as I mentioned, override some local custom variables. So I've created this value style local file, which is copied value style with certain things overridden. And in fact, actually, it would have been slightly nice if I'd just taken out everything except what is overridden there. It would have been better. For local dev, I've got some examples SQL in here, obviously the point of those are local dev.

And the people are in [INAUDIBLE] I can talk about some of the ways that you can manage SQL's production. And other than that, it's all relative to standard minimum music is going to keep local. That's the resolve that's in my etc host file that resolve to the IP address of the M, and given the arches project that would be inserted.
OK, I'm going to show a little bit about how we define what each of the processes and environment things look like. Any questions on that so far though before I go on to the next step? There we're good.
So to fire this up, I've got a rather ugly looking script here, which is a little bit more readable, and what this does is, we get about equivalent of project name, so Arches dash [INAUDIBLE] project, and that's going to be what we call our deployment and our release.
Gunny's definitions from the arches project subfolder, we're going to use the default values from that build single file, so that will override-- that's what we'll get pulled in for template, and we're going to override the image names, and actually for doing local dev this should be enough.
So if we keep running we build our local arches, and there there's some specific requirements that we can actually run this without changing anything this repository at all for our brand new custom arches, but as long as we've got those two, because [INAUDIBLE] are two images [INAUDIBLE].
And they're not going to be run with arches project YAML and still functions on the local. That well give us a series of processes, where they should start.
So that's set up our Django sets up our static in the end of the event, the [INAUDIBLE] where-- so I was having a few issues. Well, just before I do anything too dramatic to prove that this does in fact, set up an arches project, that's not just project.
Like I said, I just got three quickstart so everything should be pretty decent. And I don't think I've printed them out but this is where everything stops working.
Well, thank goodness. But I've not considered black box access this one thing. Again the and what [INAUDIBLE]. This is all based on a brand new project. You've now gone through that the project creation step. Yes. So that's-- I probably won't do that. That actually expanding their site so oops, so this RTOS trial, what I did there was going through these steps here. And then I stopped at this point.
So creating my project, and then everything and then just went from-- OK so that's not giving me this my project folder here. I'll push this, I don't think about done this yet, but that's-- well, that's not that helpful, is it? That's what it's showing me here.
I'm just assuming that it would be relatively trivial to add a container of an existing project to this. Yeah, exactly. So if you-- so in theory if you've got the project files, you'll need to files that I'm not quite sure what they said. Oh, I see OK.
Like the Docker composer or a Docker file for a project. Yes.
It's like that in here?
Yeah, and in theory-- I mean, yeah. So it's just like pretty sensible. There's nothing in any of these other modifications there, they're certainly modifications from the default setup.
Actually [INAUDIBLE] to me. Right, OK, yes. That's the key settings local a has also changed yeah. OK, so it's those plus settings local. And these ones here are obviously outside of the project, and the budget neutral. So of those can be pulled in standard dependencies in your system. But the only parameter to those at the moment critical is the actual project name.
And then on a slightly different tack, how are you identify and moving the static files into your static container? Yes. So that's-- this built Docker, first thing it does is it does doctor who is building we've got

boats and that's what this is a container, which is basically the normal dynamic container with that's my project directly thrown into it and [INAUDIBLE].
It does a build against this Docker file static. It's a little bit-- Well, it's a multi-stage build, so everything up to here is kind of pretending to this [INAUDIBLE] be up this event, but it's kind of pretending to arches that it's running in a real container briefly, and then running collect static, so gives it [INAUDIBLE] values [INAUDIBLE] random [INAUDIBLE], but it runs collect static.
So the copies are just too static containers, and runs static in the container. And then-- Yeah. Doesn't matter finishing, the arches install, because you're only interested in the static.
Exactly so--
OK.
So up from here to here is running in the normal Django container that would just go with the filter then this last bit is saying, OK, now I've got an NGINX container, I want to grab everything from that's just been built in that other container and stick it into [INAUDIBLE] and then the container here up to here just goes to them. OK.
Yeah. So that then needs [INAUDIBLE] app, so I'm going to [INAUDIBLE] something a little bit into the next segment. But I think this is probably going to help stimulate some of that discussion. Does anyone have any other questions up to that point? Because I'm going to dive into some YAML files now. No? All right.
So I go Arches project. Showing you the chart file, showing you the value to file what's in the templates. So templates-- there are a couple of things that we've set up, that are our processes or kind of like our resources within Kubernetes, same as if we were deploying [INAUDIBLE] environments.
We'll have to set up storage or set up processes and people adds in order to get [INAUDIBLE] the face of the planet. We also have some we want to be able to inject a certain configuration that we might have to find in the value file, and we also potentially want to add in-- [INAUDIBLE] nationals.
Making sure if I-- [INAUDIBLE] so yeah we do need to override the arches units file so it works all right in the Setup, and this is a good example of how we can override a file and equally, this is all template.
So if we wanted to say, OK, we want to have an environment specific variable that decides whether or not the Tiger extensions get out of the postGIS house or something like that then we can say OK, this file here will have [INAUDIBLE] opportunity to include some text and then as far as the container is concerned, it just sees this as it sees like best template of this in its file system as opposed was the original file.
So we can magically inject files on the right network. And--
Just one thought about that specific file. This is one more general comment, I think that is a part of the archers installation that we really could improve.
OK.
[INAUDIBLE] stress, the create database, like initialised database step, I think has been a little mired for a while and-- well, we kind of tried to refactor it out a bit into that file that you were just showing the override of but for example I don't think postGIS template is actually ever needed or used.
There's just pieces of that I think that we could do better to pull that out, and this is a more general comment, not about your setup here, but anything-- I guess I'm just thinking like you're doing some deployments that reference that file, if you see other ways that could be reconfigured, because I think that's a little bit of a weak spot.

One aspect of it being a weak spot is that it relies-- that setup to be relied on having a super user for a very long time. And so we didn't have it we didn't have the paradigm of someone else with super user credentials will prepare the database and then the arches user will use the database.
It was all everything assumes super user postgres credentials which in your context seems like a default anyways with the containers and maybe that's just fine but I do think that refactoring that bit of SQL out a little even further might be helpful just.
Yeah and actually there's a broader point drawing credentials that I think is really, really good to touch on. I mean like I said we're kind of coming at this from maybe working with postgres in different contexts and trying to sort of going from a particular place and see what works for us to do with it.
But yeah I mean if this creates opportunities to do things in slightly different ways then great. I mean we're with the credentials for simplicity again this is one of the things that kind of thinking for a kind of getting started with Kubernetes being able to fire that up once that postgres potential makes it easier to put it on too many instances to do some experimentation.
But then in practice setting, we'll strip that out. There is quite a nice kind of paradigm in Kubernetes for doing that's type of basically giving what called an init container that could potentially have other bit of privileges but it exits as soon as it's done it's initialization and then the actual routing containers and will be given reduced creds.
And that's the sort of thing that can be quite hard to do [INAUDIBLE] but that Kubernetes have the necessary extensions to [INAUDIBLE].
[INAUDIBLE] yeah.
And actually that's a point [INAUDIBLE] again is to try and get the simple we've just [INAUDIBLE] container and it runs but doing things like here's the initialization checks that are maybe things that are currently an entry point.
Set up to be, et cetera. If there was a [INAUDIBLE] into intercept a container that can actually even look at things like, well, what dependencies are needed for initialization? When does that initialization need to run, and when can it be skipped, and that sort of thing.
And.
Cool.
Yeah. [INAUDIBLE]. So in terms of the environment variables these come in through this configmap-env.yaml that I've added. So all of these names are just conventional they could be anything really.

And you'll see that familiar list again of [INAUDIBLE] variables. This time it's finding YAML kind of some Kubernetes formulas that we've got this API version what type of object we're looking at and the amount of data.
But you'll notice there's lots of double curly braces, and that's essentially templating indicator. So these lines here for example go in for [INAUDIBLE] 5984 is always couchdb port [INAUDIBLE] and I'm just saying [INAUDIBLE] something more flexible but with the double curly braces, we can see that actually the couchdb host is tabulated.
So rather than putting that in the bottom will run the ghost syntax because everyone likes around a bit of ghost syntax and we will state the name of the helm release in there.

So that will come out as arches-my-project-couchdb. And just to prove that I'm not making that up I should be able to [INAUDIBLE] and there we go. So that's the actual version retrieved from Kubernetes once it's been called.
So you see there in deed there is a bottom. That is templated, as I described. So it's a good version good diversion. I know all of those can be complicated and quite complex things with templates as well.
OK. So some of the other things that maybe we're highlighting here. Ingress well it's a bit of templating mass so don't look carefully at that get your eyes confused.
Essentially what this will do is take on you can see lots of values, dots that pull out what we put into our values dash local [INAUDIBLE] local environment [INAUDIBLE].
And it will turn it into a valid Kubernetes in gres before pushing it up onto the cluster. And then in reality that looks a bit less messy now [INAUDIBLE]. So actually the whole thing really boils down to this what that says is we've got host called minikube local I just [INAUDIBLE] minikube local.
And it redirects anything on basically anything at all goes to the Django and port/container and anything under /media and it overrides and sends it to the static container instead.
OK. I can go through more of those what actually went on maybe just to show is the worker. Again this looks like a bit of a mess but key thing to note is that we can override the command. So that's another copy of the actual HTTP Django container but overriding the launch command to say entry point run celery instead.
So we know that's the environment variables is got some secrets. You can see that we're pulling in sequence here instead of having the hardcoded they being pulled in from Kubernetes secrets.
So [INAUDIBLE] but those are coming in environment variables just like any other environment variable when it's running. It will see those, but those can stored in a vault or a K vault in Azure or [INAUDIBLE] or something like that equally.
And then we can question whatever other entry point processes [INAUDIBLE] of this kind of take copies of that. There's a bigger discussion about high maintenance tasks or especially non scheduled ones that's worth highlighting but probably a deep one to drive into.
OK I'm conscious that I've gone over a bit here. Has anyone got any questions on that so far or anything they want to look more at?
OK so like to gosh where did I get that [INAUDIBLE]. So I'd like to try and do a little bit of exercise around how we can look at, for example my suggestions was that we want to maybe have a stream like this that someone can get into and get something up and running real quickly first and then take that and say OK right.
The production version so we basically got both sitting on get and artifact help, which is the source of charts not for helm so people can install one to get a local version up and running but also if they want to kind of have a recommended community production set up.
Using this we can have suggested resources, amount of CPU, capacity amount of memory capacity all defined within the defaults so people can write that if they want to or not and things like scalable storage. I've done a four hour online version that set up to handle uploads and things like that and storage of the cluster is probably the entry level of that. But we can look at things like scaleable storage [INAUDIBLE].
Maybe to try and make the most of this time what are going to do is put in a link to mirror. Now I am not the company mirror expert. So I am going to have to-- I'm more used to being pulled into mirror Pod so apologies if this positioning [INAUDIBLE] rough edges of [INAUDIBLE] have been set up here.

OK. Going to stick a link in here. I'll do a quick intro to mirror as quickly as possible [INAUDIBLE]. Just out of curiosity can people raise their hand if they used mirror before please. [INAUDIBLE]

Just a little bit I've used it a tiny bit.
OK cool. OK so we've got [INAUDIBLE] raising his hand in front of me here I was beginning to wonder that given that we use it every day so and [INAUDIBLE].
OK hopefully people should've seen the link in the chat there. [INAUDIBLE] people on here.
[TYPING]

[INAUDIBLE]

[TYPING]

[INAUDIBLE]

[TYPING]

OK I'm going to do an exercise here we're going don't do some going to I do quite similar task in two different ways and just help us get an idea of-- and help us prioritize on the technical side some of the things that would be good to look at.
Could spend maybe lets say start with five minutes and see how we get on. You'll see there is poster You can create your own on the left. You should be able to feel free to have a go. You should be able to click and drag a poster and move it around. And yeah there you go you can see them kind of looking back and forth.
Feel free to create your own poster on the left hand side there's really sticky notes on the side menu side by menu. And what I'm going to suggest that we do maybe screen share the considerations and just get people to add notes either things that you thought of things that you can see in that list of considerations on future steps just given time I'm going to suggest together.
Let me just screenshot here again. Zoom is getting a wee bit stressed. Get this out of the way. OK. So yeah so things that people are particularly-- anything that you're particularly interested in either from that list or that you thought of yourself that kind of sticks out in your mind and where you sort of put it in that opportunities, basics, ideas.
On that division is just kind of-- it's very rough if someone's put something in feel free to put it again in another box, something might be looked at in a couple of different ways.
Basics being things that we really need to do to even get it up running locally well or actually on production. So we can do an exercise in second just to try and work out what needs to be done where.
Opportunities, things like what opportunities that we get from running on this in Kubernetes that might be tougher to get anywhere without it. So ideas things that could be maybe a little further right there maybe some of those things like multi-tenancy and so forth and then dependencies.
So things that are maybe not core Arches but that we sort of need to consider [INAUDIBLE] to be set up or how we can [INAUDIBLE], things like that.

OK I'm going to set a timer. Discovered that you can now pick music I'm going to save you all doing that as we had some painful experiences selecting those. Actually if you had a bit of music in [INAUDIBLE] self.
Is everyone very clear on how that works. Any questions on that? You all seem to be getting into it so that's good to know.
Phil where do you-- which category do you need to put things like things you have to consider that might be not so much problematic but that might kind of throw a spanner in the works as it were?
I'd say [INAUDIBLE].
OK.
If yeah if it's something that would be a potential blocker.
OK brilliant cheers.
[TYPING]

You're free to create more posters to help us.
[TYPING]

OK cool. OK that's quite a collection that we've got there. So 44. Yeah we're OK. So is there any that people want to talk about particularly or kind of jumped out to them there or they want to explain in more detail or you have enough .
I can say basics and essentials creating a new project will set up a composed and community basics for that project is definitely something we discussed internally at Farallon.
Yeah.
As we have created the templates for creating a project with-- and as we worked on Docker recently we've talked about adding Docker compose and Docker files to our project templates by default.
Oh right yeah that will be handy. Yeah and I can see how that's once that's there then it is essentially Kubernetes but if it assumes that that's the Docker set up or Docker compose set up we should be able to set up Kubernetes config that you can just fire it up with no additional work.
Standardize that.
Yeah because I think with Kubernetes if we can ease the path on to it because it can be [INAUDIBLE] something and it's kind of like you've got your Docker compose running. OK it's three commands and now your project as you see it Docker compose is not running on your minikube and once you've done that well, at least you've got some confidence that you can go on [INAUDIBLE] yeah.
OK I'm developer out of water here but I'm going to do adult food. What I'm going to do is see if I can, that's right. [INAUDIBLE] yes that's it. OK
[TYPING]

So I'm going to do a three minutes Well maybe we could five minutes just because you haven't been through this, we're going to have to do a couple [INAUDIBLE] OK.
So if click vote now we should all find that you've got an invitation to vote and when you click it, then you should see what's actually started you should see that all those sticky notes appear with little pluses beside them.

And if you can vote for the ones that you think are critical for being able to just start playing around with this locally usefully genuinely usefully kind of the sort of thing that we've been showing. So the 10 that you'd really say looking over those these are things that I really want to have working in my local dev. You can vote multiple times for the same one if you really, really like it and that's only positive [INAUDIBLE].
Now it doesn't need to be the kind of basic essentials ones necessarily like it something that you think really should work locally even if it's kind of a bonus production. Am afraid to put [INAUDIBLE] on.
Did that work forever or did I just. Is everyone else stuff loading? Yeah. Right yeah [INAUDIBLE].
OK I actually can't see this [INAUDIBLE] and so [INAUDIBLE] and so forth there's 26 seconds left so if you haven't the next 26 seconds. [INAUDIBLE] time here. 5 seconds left and there you go. OK so you always having to think very hard about this OK cool.
I think I might screen share this unless can you see the results [INAUDIBLE] You can see the results. OK. Great. So we've got at the top there create a new project we've set up so that that's been mentioned [INAUDIBLE].
So I'm going to pull this out [INAUDIBLE] OK so I'm going to pull out the top ones here. So I've got migrations. I've got five there. Standardization of creating when then we set up and I guess that's both instructions and [INAUDIBLE] that these are on the plan line. Standardized monitoring it's interesting one OK. [INAUDIBLE] standardized monitoring.
OK, quick [INAUDIBLE] set up when it's there and maybe stop at three that's given us quite a few minutes to kind of hungover screenshot because I was so suspicious [INAUDIBLE] wrong so at least that way i can [INAUDIBLE] the reference. That's when there's a three default values required [INAUDIBLE].
OK. So [INAUDIBLE] that means I can then that was to explain standardized monitoring [INAUDIBLE] to this migrations [INAUDIBLE] working.
Point out that two of those the two of them are very similar. They're running Django commands and how to run a command line data loading.
Yes, the [INAUDIBLE].
Same with the integration with cloud platforms in dependencies.
Yeah.
Looks like those two are similar.
That's a good point actually. [INAUDIBLE]. [INAUDIBLE] what's the other one the [INAUDIBLE]

It's all right for you [INAUDIBLE] in the morning but my parents already had to do a day here so I'm having to look and try and see and yet [INAUDIBLE]. Oh, yes. I see. Sorry. I got you. [INAUDIBLE]
Yeah OK. So [INAUDIBLE].
So what I'm going to do is kind of put the ones there for my priorities for dev so that's OK. Rechart that. What I like to do is now use a shorter one.
[TYPING]

And with the remaining ones if we can do [INAUDIBLE] do a prioritization for production. So basically we are going to give seven boats this time things that would be a priority to get working before going into production.

So well that's gone and I'll give three minutes because now we've seen that so [INAUDIBLE] we'll have a rough idea of what once didn't get pulled in that [INAUDIBLE]. So OK that's [INAUDIBLE]. Don't be like me and accidentally vote for enable.

I seem to be having problems it's only allowing it's kind of clicking it to highlight it to edit rather than allowing me to vote so.
Yeah, [INAUDIBLE].
I was having the same issue I just reloaded the page and it seems to have fixed it.
OK brilliant thank you.
[INAUDIBLE] just saying thank you for sorting that. I also realized I'm not entirely sure I extend the voting times [INAUDIBLE] We're coming down 10 seconds. Is everyone able to-- is that working OK for you now, Chantelle?
Yes.
Yeah.
One screen. One screen.
Three seconds and you need to answer the question as well so.
Yeah.
I'll try to buy [INAUDIBLE].
OK did everyone manage to get a chance to vote? [INAUDIBLE] right. Results are OK. Well interesting. To realize it's six so if anyone needs to drop [INAUDIBLE]. I'll try and finish up as quickly as possible here.
I think one of my question is that I'd underestimate the time it would take us to go through this, so I apologize. But was keen just trying to give us a two hour window to get through things as successfully as possible. In that way but there still may be help draw some things together and then set up essentially a second workshop to dive into specific aspects of this.
OK so everyone can see the results yeah. So again there's some of those pairs if I'm missing a pair of things that are essentially seen in the votes. Just do tell me. It's scaling according to each department. It's top there.
We've got a couple of forms of self-healing inside it and these are backing up. I think that integration a great platform to deliver funds for different platforms I think that that was one that was mentioned there and going to put this together call out a four [INAUDIBLE]

And I think that's what we get [INAUDIBLE]. So let's take this [INAUDIBLE] in as well and OK. And and. OK I'm going to do one last one I think most are still on here. So we're going to do this one for two minutes I'm going to give you these four votes each.
And this is to pick any of the rest that you would like to see discussed early in other words that you would be particularly interested in the Kubernetes discussion if these were included.
So click up to four [INAUDIBLE] times [INAUDIBLE]. And it doesn't matter whether it's critical deployment or not just things that are real pictures that you would want to [INAUDIBLE]

So two minutes up here and there we go. And I'm just the ones in the squares or the ones that we pulled out.

So I think one of the things that I'm hoping has happened here and I think it has is that some of the financial complexities for doing production deployment that we've actually kind of managed to highlight collectively.
Obviously, as anything else comes up as we start going through this, this just gives us a kind of shopping list of things to factor them and start the discussion around.
[INAUDIBLE]

Seven seconds. OK. OK.
[INAUDIBLE]

So over here we've got basic mapping and map search and then and in [INAUDIBLE]. OK, ability to have cross instances. I'll be really interested if there's any work being done on that. Obviously, if there hasn't, I think [INAUDIBLE]. Lou Donaldson [INAUDIBLE] creation, OK.
And then we've got the other ones here just to highlight them [INAUDIBLE] most generic term I can think. [INAUDIBLE] dimensions, pages, standing with apps.
OK so again that's a little bit like that [INAUDIBLE] collector getting local tendency set up, and then local tendency.
So actually we got those actualised instant management. So there's a few those that are actually kind add together [INAUDIBLE]. Yeah assets and bonds. OK and that's really gives us a bit of a framework to start working through these [INAUDIBLE].
So thank you very much. I don't want to [INAUDIBLE] make sure [INAUDIBLE]. What I'll do is try and pull this together in a little bit of a clearer form and then it's [INAUDIBLE].
The feedback would be good here. My thinking is that we could maybe do another session looking at one part of it looking at the development pieces there or at least the highest priority ones. One part looking at the production pieces there.
And I'll do another two hour session because I think that's probably a window that works for people and then actually just go into the practical [INAUDIBLE] steps of what needs done or if maybe we can make decisions that are [INAUDIBLE].
Does that sound good to people does that sound like a reasonable next step?
Yeah that sounds good to me.
It sounds good to me.
Perfect.
One thing is I don't know how many people outside of [INAUDIBLE] are running Kubernetes I know that I'm not right now. I don't know if that would be worth a session to get that up and running or if it's something I just do on my own. I would have to dive into it and see but I don't know if that would be valuable as well.
Yeah, yeah I'm more than happy to see that. I mean actually what I could do is trying to [INAUDIBLE] is to take the example that we've got. So I obviously I want to keep this one time. But what you can do is I do a complete tear down and just set that up and get to that login screen and menu from my local VM.
Would that be helpful or would it be considerably more useful to just do a bit more and try and get on to [INAUDIBLE] or would local be a good starting point?

I think local would be a good starting point I can't speak for everybody obviously but for me I think that be good.
I think most of our development when we first start out with things is done kind of locally and then we push it up to our Azure setup. So yeah I think sort of just getting our heads around how it works I agree with a local getting a local kind of set up thought it would be really helpful.
Yeah, that makes sense. I mean not that-- I think that's a fair point is generally what we would do that ourselves. I think where we sometimes find with Kubernetes is that you end up with a slightly different set of locally than you would on the cloud because maybe keeping some local kind of things.
Like you've got one node or stuff like that but you don't have to think about some thinking about it in a way but I think that's a good starting point if people are happy with that, that would be good.
OK I will schedule that any feedback on time of day. Is that working for people? I realize it's early and late both, but--
Yeah this time works for me and on Tuesdays as well I think it's probably the easiest time of the week or day of the week.
So potentially if we said doing something like this in well I think the Arches UK meetup is coming up very shortly and so maybe just the far side of that all right
Thanks very much folks. Really appreciate you joining and I hope that was interesting and you have lots of fun with me which is added surprise bonus. And yeah we'll look forward to picking up.
[INAUDIBLE] thank you for running this.
Thank you very much for [INAUDIBLE]. Appreciate it.
Yeah, no worries. Cool. Talk to y'all soon.
All right.
OK, bye.
All right