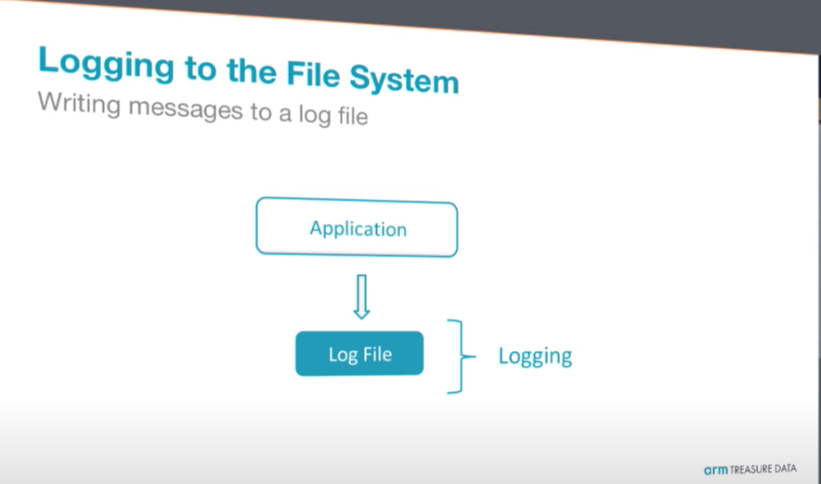
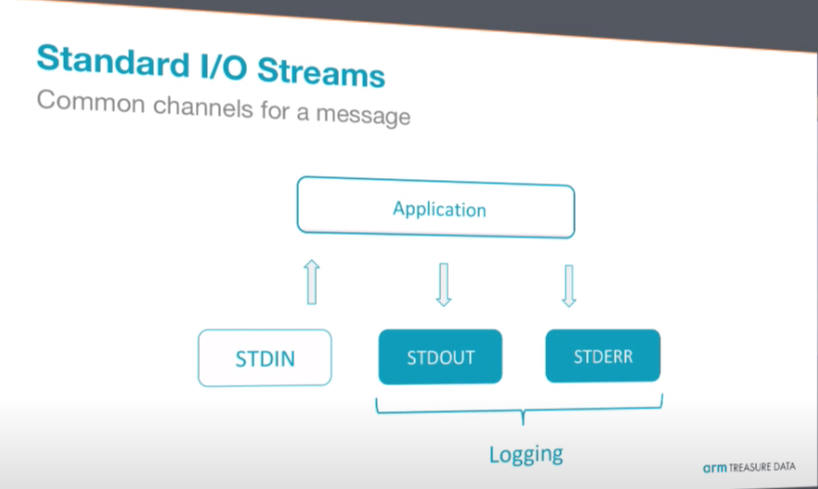
**Logging context in kubernetes**

Before talking about a logging in kubernetes itself we need to understand how applications works with logging. The simplest logging is that when one application triggers a message and that message goes to a file and that becomes logging

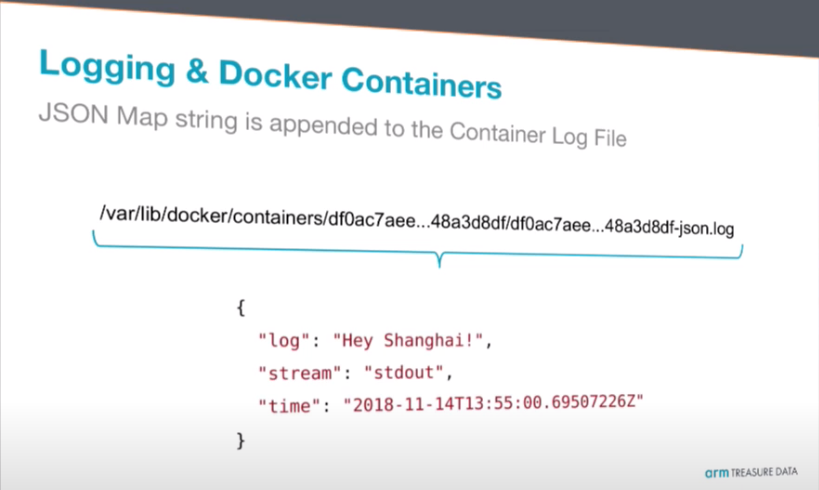


if you are familiar with UNIX or Linux, in general you know that every program has at least three stream interface, standard input the standard output and standard error and this interface are really important for the container

so login happen mostly here in the standard output and standard error so that is one way that applications can start sending some messages to the outside saying something is happening with my application and maybe something is missing or something is wrong okay now how this works in log

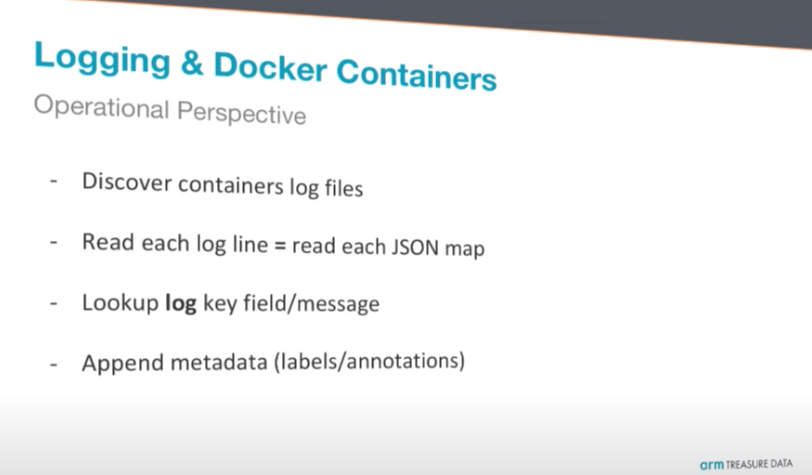


Now how this logging works with docker containers or with containers in general your application triggers a message like "hey Shanghai" but that message is not just the "hey Shanghai" right it has a timestamp a time when this message was created, stream from where this message was created and time it was created, and that message is stored in the file system located at /var/lib/docker/containers/<conatinerID>/<conatinerID>-json.log



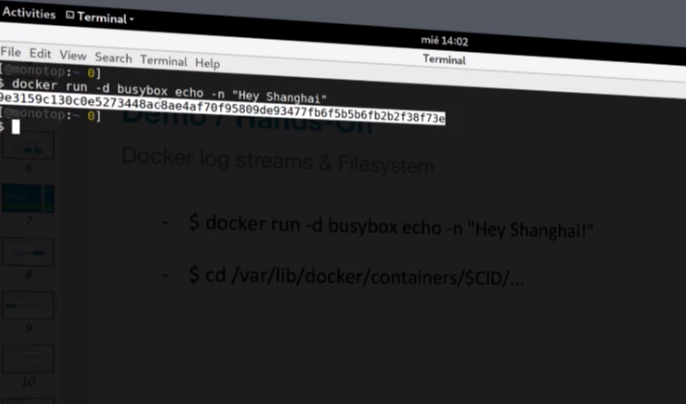
Of course if you want to do some application troubleshooting I'm sure that you don't want to go to the file system and look for that file because that is a pain, if you have a hundred of containers you have a hundred of different log files with different hashes and you don't know what that mean in general

when working with logging you have to understand that if one application generates a message that message go to the file system, you need to understand where these logs are stored and how to read them and look up for the specific fields inside the JSON map to get real message, of course sometimes you would like to append some metadata to that message.



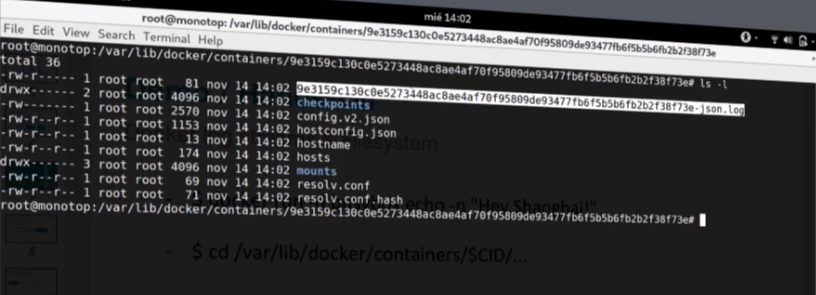
A very quick hands-on here to understand how this works of course this is not a way to do logging, but I am showing for demo purpose.

For example, I'm going to run a simple container which is called busy box with a message Hey Shanghai

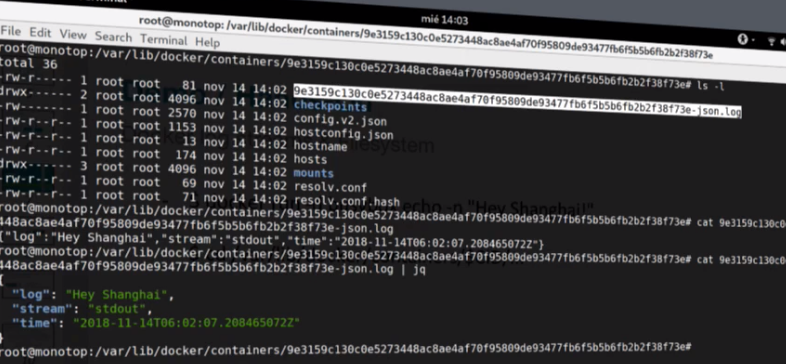


We got here is a container ID this is what happening behind the scenes actually you don't need to care about this but it's good to understand how it works because you can improve that in the top level.

If you got to file location /var/lib/docker/containers/<conatinerID>/ you see there is log file



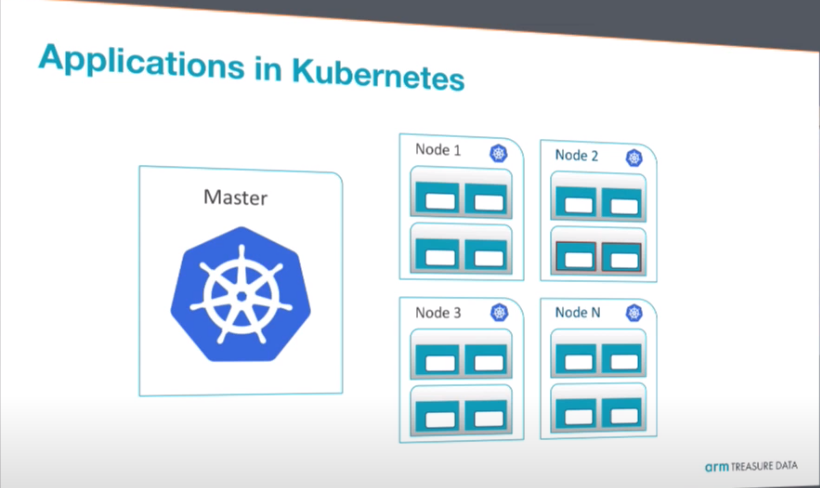
Then if you see what that file contain is



okay I took it a message on the command line and that message was stored in the file system in a JSON map and then you got it there but of course you don't want to do that manually okay so how this is correlated with a kubernetes

Before correlating this logging with kubernetes 1st we need to understand how kubernetes works

You have one application or at least from your perspective you want to deploy one application, this application will be deployed in a container that container is inside a concept which is called a pod and a pod can have many containers which means multiple applications, and in an kubernetes node which can be a bare metal machine where you have many pods and a cluster is composed by many nodes along with control plane and node components like api server, scheduler, kubeproxy etc.

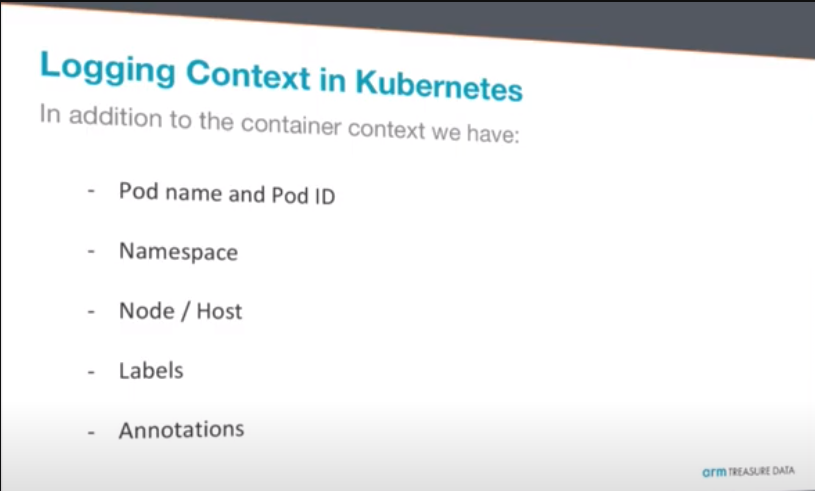


if you think about this if you are going to do logging on this you have logs from this container which belongs to this pod belongs to this node on this cluster so it's not so simple it’s not just container id and my log

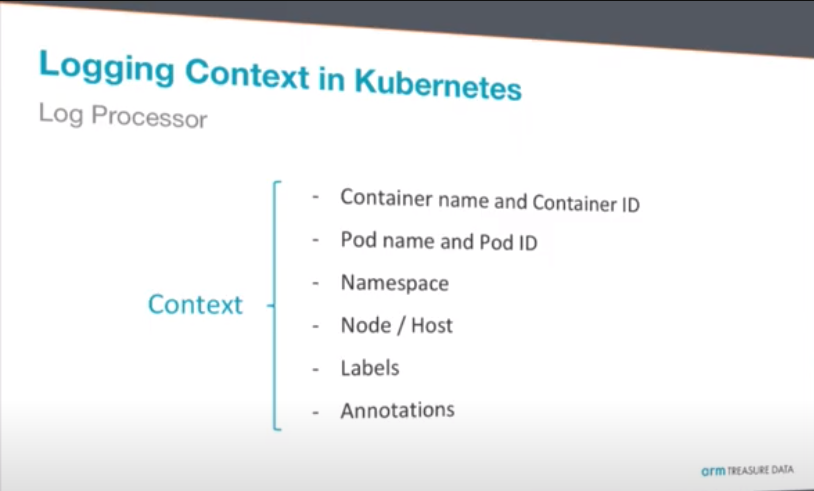
**Logging context in kubernetes**

I said we have the application we have the containers we have the pods and the node and each one of them has a lot of data that allows us to understand a what is the context of the application that is running

From a log message you want to understand what is the pod name, what is the namespace, what is the host name what are the labels in kubernetes and annotations?

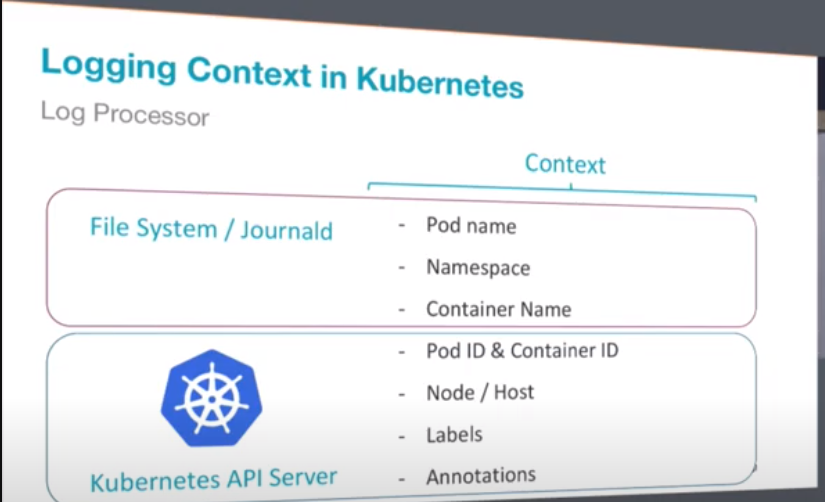
****

So as a context we take all this information the container name, container ID, namespace, node and so on and that is something that needs to be solved by a login processor because you don't want to do that manually it will be a pain.



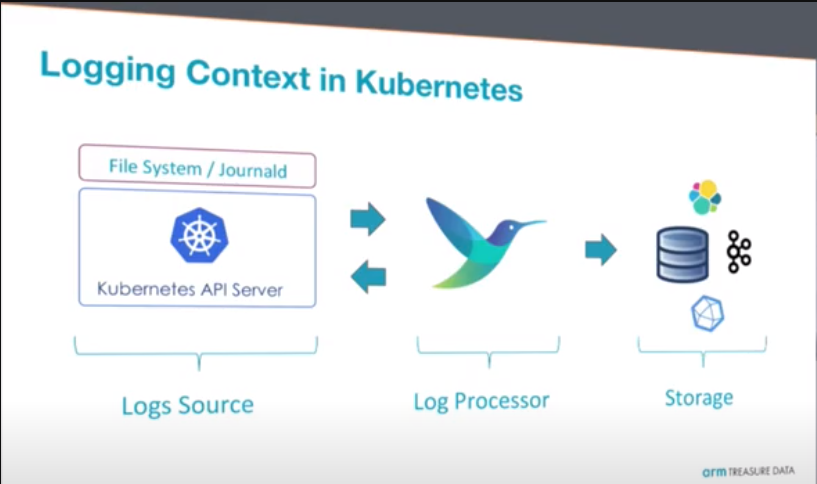
from where I'm going to get this context for the application that is generating some log files? Some part of context like pod name the namespace and the container name comes from the file system where it's running that container.

They are more information that is outside of the context of that node and that information is inside the kubernetes master or API server.



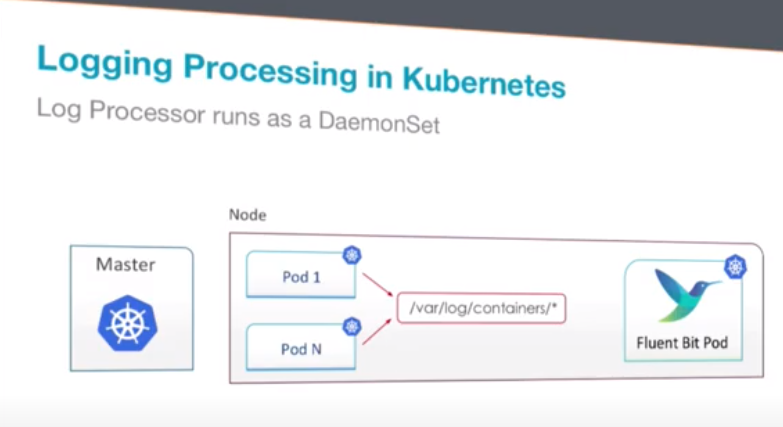
Because logging is just to collect data and aggregate data in one central place to do that you need tool called log processor

so when you have the log processor working inside your kubernetes cluster then you go to your real aim that is aggregate and store that information in one central place your data base a cloud service or whatever you're using for your own needs



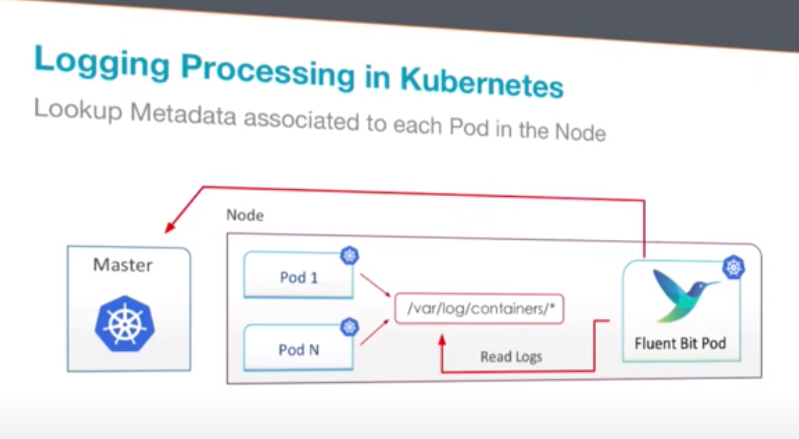
**Log Processing in kubernetes**

As we see below there is node which contain two pods or two applications and these applications generating log those logs are trapped by container engine and being stored in file system this is one way, some kubernetes cluster uses systemd with journald. Log processor run in every node as daemonset

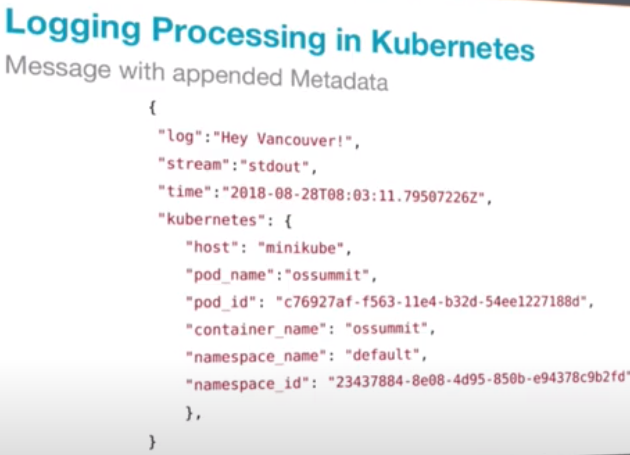
****

These log processor start consuming logs that are being stored on that node just one part of the context right… we need to get more information and that information come from api server

So first log processor will go to file system first and then go to the API server and merge all that information and then deliver that look outside to the database.



Now the simple message we generated before appended with time stamp and streem now become like this and with this you will get real context



Now with the help of elastic search you can query the logs belongs to specific pod