

## Open Source Week - Day 3

IBM Code Tech Talks

02/28/2018

<https://developer.ibm.com/code/techtalks/open-source-week/>

>> Hear from the open source team at IBM, and we are going to kick off this afternoon's session, this afternoon in the East Coast anyway, with an update on Cloud Foundry from Michael Maximilien. Max is a research staff member at IBM and he is one of our leaders at in the Cloud Foundry project at the Cloud Foundry Foundation. Take it away.

>> MICHAEL MAXIMILIEN: Thank you, Chris. I think on the website it reads Michael Maximilien, Chris mentioned Mike, Max, I always go by Dr. Max, this is just to make your life hard so you cannot find, so just call me Max.

What I want to do is cover as Chris mentioned Cloud Foundry and it is going to be similar to what we have been doing for each one of the other technologies that we are focusing on.

The agenda, covering the following different categories, what is Cloud Foundry, this is origin story where was it created, where did it start. Of course let's talk about the current community of Cloud Foundry where it is right now and who are involved. Then we will go into a technical overview, it's a very complex project, like most of the other presentations that you will see. We are only covering the surface. But hopefully we will give you enough details that you can go and find implementation yourself. Then we will cover the current status of Cloud Foundry which I'm well aware of since I'm part of the community actively.

Then we will talk about what is coming next so that way you can get a view of the future. We will mention to you how IBM is using Cloud Foundry and then give you ideas on how to get started, and of course some links for you to actually take action right now.

With that, let's get going. We have 30 minutes. There is a lot of to cover. Hopefully we can have questions as well. What is Cloud Foundry? The easiest way to understand it, for me, is think of it as an operating system for the Cloud. It has very specific things that it does for the Cloud. This diagram on the right covers key aspects of Cloud Foundry. Generally think of it as a operating system for the Cloud. Whether or not you use

operating system for your Windows machine or your Mac or portable device, you need an [inaudible] manage the software, hardware for you to you can run software. It's the same thing. Cloud Foundry does that for you except it's targeting a Cloud somewhere in the world.

And in the application that it's managing for you are the application that you bring to the platform. Multi tenant, you have multiple users, eventually in any particular installation of Cloud Foundry. As I mentioned the diagram on the right gives you the three main aspects. What are the features of this operating system for the Cloud. It is Cloud agnostic. You can pick up foundry as is, there is a shim which we will talk about later, that you have to change, basically it's called a CPI. You have to get the correct CPI for the Cloud you are going to target. You can run Cloud Foundry on Cloud like SoftLayer, AWS, GCP as OpenStack, vSphere and many more. There is over twelve production installation of Cloud Foundry as we speak right now, that people are using.

At IBM we use it on SoftLayer and vSphere and a few others. That is the first dimension. The second one is that it's framework and language agnostic. That means you can bring any application to Cloud Foundry and make it work, in application and run them and scale them and so on. Those application can be a variety of different framework and languages. For instance, you can have go lang application, Java application, Node.js application and even esoteric languages like elixir, you can run them, Cloud Foundry has support for Swift within the first couple days of it being released. Is it easy for us to add support for new framework and new languages. The beauty is you treat all the application whatever language the same way.

The last part of how to understand Cloud Foundry is that it also exposes a marketplace of services. When you run your application you need to connect it to some existing service, for instance, database where you can receive your data, service can be a lot more [inaudible] Blockchain, they have a bunch of services they are building for Blockchain. You will be able to use your application using those services, the same thing for any variety of different external components that you want to bring into your application or you want your application to make use of.

That open marketplace of service is open obviously, but what I mean to say is that it actually was extracted out of Cloud Foundry and became the open service broker that now you can use in Kubernetes and Docker and any other Cloud platforms.

In my opinion, it's the most established PAAS on the market. Not just a opinion, the fact is it is in terms of number of deployment, number of customers using it in production and

number of offerings, so IBM obviously has a big offering, pivotal G.E., SAP and many more have offerings all over the world. So whatever you can get in Cloud Foundry. How is it different, that is the important thing to mention, is that it's opinionated.

IBM, we have a big IBM Cloud that has multiple components. You can think of it as the application service for the IBM Cloud. But what do I mean by opinionated quickly is that it focuses on productivity of the developers.

We don't try to make you learn jargon that you don't know that you don't use. It is very much directly meant for you to bring your application, focus your app, when you are ready, you push it, when you are ready to scale you issue one command to scale it. When you are ready to connect it to a microservice you issue another command to connect it. It is very much targeted towards developers and also enterprises, because it has all the features for an enterprise to manage, not only the installation but also to create organizations, access control and so on.

I like to think of it as an operating system for the Cloud the same way that a Mac OS is a operating system for an individual laptop. You contrast this with something like Kubernetes, for instance, which is also a operating system for the Cloud but more at the level of what Linux will like meaning that it's low level and granular. You would build a shell on top. Cloud Foundry is all of that. It provides the low level layers that are specifically hidden from you and it provides a shell to make it easy.

That is what Cloud Foundry is.

How was it created? It was started by a ex Googler, Darik Collision at IBM in 2009. In some ways, Derek and his team took ideas that they had seen at Google and recreated it. So some of the things that, I mean I don't know the secrets of how they do things at Google, but I can tell you that when you discuss Cloud Foundry with different people that are aware of both, they can see similarities.

Obviously things have changed a lot since, one of the things that happened is that Cloud Foundry as a VMWare kind of merged with EMC. It got extracted and folded into pivotal to fix issues that were happening with it at the time. In 2013, Cloud Foundry became mostly based out of pivotal but it still remain open source, so IBM was very much involved. During that time, of course, as a platform that is trying to get into production, that is trying to provide useful software for the Cloud, it went into some I guess you could say growing pains. One of the important problems was that it did not have an open governance model.

I know at the time Chris was working with us on this, and he pushed quite hard and the rest of the team at IBM to make them adopt a open governance model. That did happen in 2015. In 2015, the Cloud Foundry Foundation was created. It's a classic, not-for-profit 501C under the Linux Foundation. With that came a very open way to govern the source code. And also as a consequence, you could see a huge amount of increase in terms of membership in Cloud Foundry.

The diagram on the right shows you a little bit of the current members of the last summit, specifically show the platinum members, gold, silver members. Last year as a notable year, Google and Microsoft joined the Cloud Foundry Foundation. The foundation keeps growing. I'm sure this year in Boston we are going to announce even more. Look for that. Go to the website. You can check to see the current number of members of the foundation.

How is the community? This is open source. It is great to have a governance model that is open, but how is it organized? Is it easy for you to find your way around Cloud Foundry. It is not too hard. It's a very big project with lots of GitHub individual projects. But there is a method to the madness. The way to think of it is there are three different categories of how you can organize Cloud Foundry. The first one is that there is the organization GitHub organization called Cloud Foundry. It is under the Linux Foundation's provision. This is where all of the core repository, the core code exists, so things like runtime is there, Cloud control, which provides APIs is there. This particular organization in terms of the technical leadership I would say, a lot of the projects are led by pivotal but there are other member organization that are part of it. Of course, we have IBMers contributing to all parts of it.

The second one is more recent organization, because projects are going, tons of people using it, what happens is that people want to extend it. Where do you put the extension before they become core projects. That is what Cloud Foundry incubator is. It is an organization in GitHub where all extensions, all interesting sort of additions to Cloud Foundry experimentation of Cloud Foundry are added.

This is going very fast, as you can imagine. There are over 30 active projects right now. We lead this organization, as a matter of fact I do lead it, this particular organization.

Then there is the third part in GitHub, you go to Cloud Foundry community, and this is where it's almost like a scratch bag but there is tons of cool stuff happening there. It's an open organization where anybody can come and contribute projects, related to Cloud Foundry and you can create your own teams and projects and test out, and eventually those can flow

into incubator. There is a process for that. I'll talk about this later. From incubation you can flow into Cloud Foundry as a code project.

This is the simplest way to think of the way the Cloud Foundry community is organized.

Let's go into a little bit of the technical overview. I have two slides on this. The first is to talk about the runtime and the second one is to talk about how you DevOp Cloud Foundry, how you manage it and what are the tools that are provided for that.

The way to think of Cloud Foundry is opinionated. It provides this platform for you that you can take your code and issue one command, CF push your code and it does it for you. Obviously there is stuff going on in the back end. First thing to think of when you look at the stack, at the top layer you have the router, routing services, so when you push your app, that app needs to have a road so that you can access it and because apps crash all the time, you need to be able to recreate them and reattach the route and so on.

That is the top area. The next one is authentication piece, where you can authenticate who you are, and do you have permission to push application, you have permission to read applications and scale applications and so on. Obviously, as I mentioned, it's geared towards enterprise. There are ways for you to create groups of apps, which we call space. You can create organization within Cloud Foundry so that within your particular company, you can have multiple organizations, so one organization for IT, for instance, one for HR and so on.

All of this has to be managed, the access control. That is what UA does. Cloud Foundry provides you a way to extend this, so the log-in server is a way for you to add your own log-in and authentication piece, so that people if you wanted to connect it to your enterprise authentication directly, you can also do so.

Below this are the things like the life cycling or lifecycle of application and services, so one project also represented here is called Cloud controller. This is the API to all of Cloud Foundry, everything under it. And next to it are services which is what I mention now extended or opened as open service broker. There is a health management piece that tries to actively keep track of application, make sure they are running okay and take action, because the system itself, when you push your app and the app crashes it doesn't say crashed, the system will restart it for you.

Underneath this is the execution piece of Cloud Foundry, ran by a project called Diego. It is a rewrite of an existing code base called DEA, Diego is a go version of this. The Diego piece is an orchestrator for containers. You hear about containers. Cloud Foundry has been using containers from the beginning.

Those all run as essentially runc containers similar to Docker or Kubernetes or any other platform. The orchestration piece of that is called Diego. There are multiple pieces to it, these are all components that allows the runtime to be able to manage those containers correctly and put the right set of software that you are pushing to it, restart them, keep them healthy and so on.

Under this there is a message bus that allows all the components to talk to each other, but these are ones that need to talk to each other. You can expose things like metrics collectors and so on. Cloud Foundry is a full stack to provide you top to bottom a solution for you to manage, provide for your Cloud, whether that Cloud be public Cloud or it is inside the enterprise. It provides you the features like log aggregation so you can access log, when your apps crash you can keep track of how many times in a organization apps are crashing.

It is more components as I mention in the extension that you can add to manage your installation of Cloud Foundry.

With that let's move to the next part of the deep dive or the technical overview, because it's a component called Bosh, people are well aware of because I'm part of that team actively contributing to it, the way to think of Bosh is as you take Cloud Foundry and you hear all the good things that it does for you, how do you manage Cloud Foundry itself? Cloud Foundry is managing your application, your service, and your users, the organization, your status and so on but how do you manage Cloud Foundry itself.

The way is using a tool set called Bosh. It was part of Cloud Foundry. Now it is a separate tool set because you can use it to deploy other things. As a matter of fact, one of the things that pivotal did recently is to manage Kubernetes with Bosh. That is one thing that you will see. You can manage anything else. But Bosh is pretty much primarily designed to manage Cloud Foundry. It is kind of deep dive into it very quickly, so I can give you the big animal picture, so the way you think of Bosh, it's a tool set to manage any kind of release software. There are some key components. One component is the software has to be packaged in a particular way, directories where you put your source code, binaries, configuration, where you put your scripts and so on. That is what we call a Bosh release. It is a specific way. We have a CLI that allows you to create a Bosh release and you can modify it.

All parts of Cloud Foundry are created as Bosh releases. The next piece is all of this has to run on top of VMs or in some containers. Those have to have an operating system. There needs to be a base operating system so that you can layer your software on top of it. We call this a stem cell. Right now we

support two or three kinds of stem cell, Ubuntu, sen OS and there is another one that we support but the stem cell is Linux. But stripped down to minimal amount of what you need. This is very important because one of the things that you know if you don't know that is a problem, but you should know that, that they are really bad actors on the Internet, that means that you have to keep your software up to date all the time. This is one way we do it. Basically whenever there is a issue on the Linux kernel, we release a stem cell the next day. This is all tested.

The whole Bosh team, part of their job is to do this. Once you have those two components, another piece is to say, I need to lay out my software. This is how I'm going to lay it out. In Cloud Foundry there is various components, they each have to belong some place. Diego has to create cells so it can run your job, how many of those you need, where do you put them, in different AZs, all of that is what we call deployment manifest.

We have tools to make it easy for you to use, where you specify where things are and important piece is how to network them. Bosh will take the manifest, the stem cell that you want to use and the release that you want to deploy and it will deploy it for you.

The best part is Bosh will not just deploy it for you and that is it. It will also help maintain it. You have a new stem cell, you can hold that stem cell, you never have to stop your installation, you can take your stem cell. When you have a new update to a release, you can deploy that. It will know where that release is deployed, which VM or which container and it will roll that container and do it in a way that your system is still alive. It's a big operation. It seems simple but trust me, it's as much good as the whole Cloud Foundry itself. We added recently in Bosh DNS where the different components can know about each other. That was a big hole in the platform. We didn't have that. We depended on the support DNS so we now have that built into the system.

What is the status of Cloud Foundry? One of the things that you don't hear about a lot because maybe the other projects are sexier is that Cloud Foundry works at 250,000 applications and more. We stopped testing it because it was costing so much money to test it.

You can rest assured that with Cloud Foundry you can scale, will run anything that exists right now. Maybe other things will get better but it's the most scalable form that we have. The second piece is we have application auto scaling. When you put your application, you can specify rules so that the application can scale based on the input request or based on some rule that you do like for instance, you can say around some

holiday you want to scale to some level. This is led by IBM. It was extracted out of BlueMix and contributed to the community.

Another piece is applications themselves, like Cloud Foundry can be ruled in a way, in a deploy way, you want to do similar things with your application so that you can deploy your version, test it out and make sure everything is okay before you roll everything so you can do that. Another piece that was released last year and that is coming soon, I think it's already there, the IBM Cloud, is isolation. By this we mean, Cloud Foundry application is isolated from other ones but what about isolating completely so that your application or organization is in a particular VM versus some other application that is running on separate VMs. You can specify that. We can isolate the network. We have added better support for networking so when you deploy microservices, they can create networks with each other and have faster access.

We have better logs which is always an issue. As you know, in anything like this, things go wrong, application crash, system component needs to change, need to be updated. How do you know on all this? It is through the logs. Having the best logs that you can is important.

As I was mentioning, the Bosh DNS is a feature that was added recently that provides a full HA/DR solution for Cloud Foundry. No longer do you need to deploy Cloud Foundry in a place where there is existing DNS. Bosh provides DNS. If something goes wrong all the nodes know about each other. They can connect. There is full-blown HA/DR, we don't depend on one particular node, even the Bosh director which is the central brain of Bosh, even if it goes down, the system can continue and you can recreate a new Bosh director, reconnect your database and everything is okay.

What is next for Cloud Foundry? This is the piece that I've been leading the past two years. I'll briefly mention but it's super important but what you get with this is showing when your platform matures what happens, people are going to put it in different direction. How do you extend it? We made it easy to extend. Cloud Foundry has three organizations, runtime, Bosh, tool set, and extensions.

Extensions are a way to think of them is innovation. We work hard to create a simple process, two pages. You read it, you understand how it works, and you can bring your extension. If there is multiple levels of control to not allow any extension to come into the platform, so one thing people do is work with me directly to contribute an extension so that I can give them peer review if they want. Otherwise, we essentially go through the process and actually get to a point where people can vote

and accept your extension or not accept it for incubation. Project that got incubated since March 17 when we started the extension PMT are Kubernetes Bosh release, which we call Cloud Foundry container runtime. You can take this release and Kubernetes and manage it like you manage Cloud Foundry.

That was worked through the extension process and is now a thing. Another one is BBR so that you can have a single place, single way to manage, to back up your entire Cloud Foundry and restore it. Each component can have customized data that they need to back up and restore.

Now you can think of Cloud Foundry installation and pretty much issue one command and back up the entire system, and then you can also restore it. The point is that there is a tool set for that. Another thing that pivotal contributed is called CF local which allows you to, in your local environment, get a similar experience like you have with Docker, where you can use your little laptop and create your application, connect it to services, do all the things that you need to do without a installation of Cloud Foundry. When you are ready, you can go to the IBM Cloud and push that code.

You can do all your things locally. Another one that got contributed recently, super fantastic, you should look at it, is a complete UI for Cloud Foundry that SUSE contributed called stratus UI. It is a complete UI in terms of Cloud Foundry like a lot of tools are driven by command line. This is a UI. It provides you with a way for to you see all your application. You can have multiple Cloud Foundry so you can connect to the IBM Cloud, connect to G.E. or to the SUSE Cloud, and then you can see all your applications. You can slice and dice them however you want. You can issue commands and so on.

It is a UI for Cloud Foundry, completely open source, SUSE contributed that. We have a pipeline of projects and we keep a catalog of that so you can look at it.

To finish, how is IBM using Cloud Foundry? As I mentioned, application service in the IBM Cloud is using Cloud Foundry. The runtime is running on Diego and a lot of our operations for services that we provide, as you know we have catalog of service that is unsurpassed. That catalog, each one of the services are managed at Bosh releases.

I would highly recommend you to look at that. We led the effort around taking the service model in Cloud Foundry and make it so that the same service can also be used in Kubernetes or Docker. We also lead the community calls, the cab calls and I lead those, a monthly cadence for discussion about extension, discussion about the PMCs and different projects, it's a way for people to be involved. If you are new to Cloud Foundry, every month we have a place for you to come and talk to all the

leaders in Cloud Foundry, and if you have a project that you want to contribute, you can do it.

How do you get started? Very easy, the easiest thing is look at the documentation. Each documentation page has a PR. If you see a error you want to contribute, you can do that. That is the first thing. You can go to the source code, tune it and make changes and contribute back. Every team has time dedicated to spend time in reviewing PRs.

In Cloud Foundry, it may be uniquely, as a unique way for you to contribute called the Dojo program. This is something that pivotal started and also other member companies have created Dojos around the world. But the idea behind the Dojo is that you can come and embed yourself with Cloud Foundry project, and for a period of four weeks to six weeks, you work at the company so you can work at pivotal. I spend time there so I work there but you can also work at IBM, G.E. and various places where you can contribute directly. The idea is that you don't expect it to come with all the knowledge, but as you spend the time, you can build the knowledge and build a relationship with the different people in the project.

The Dojo program is gated, because it's significant expense from the person coming and also the company expecting you so there is a test that you have to pass to make sure you are proficient and also that you can test drive and pair. But it is not very hard. Once you pass that test, you get invited to the different organizations. You want to target the Dojo, that makes the most sense for you.

Final thing I want to mention is that there is a way to submit extensions that I mentioned already. I have the links here. There are tons of blogs, E docs and we have two summits around the world, one in New York and one in Europe. This year it will be in Boston. We have another one in Switzerland in October.

As a call to action, I want to invite you to get involved, join the meetup. Attend a conference, if you want to meet the people there. More importantly try to do Dojo. It is one of those fantastic ways to learn, meet your peers, build relationships and also contribute by the time you are done with the Dojo, the Dojo graduate, whichever project you went, you typically end up with having commit access to the code and you can continue working. I am 55 seconds over. I apologize, Chris. I will stop here and see if there are any questions. If there is not -- I don't know if we have time for that.

>> CHRIS FERRIS: We are going to have to take the questions to the chat, if there are any.

But thanks a lot, Max. Dr. Max, as the case may be.

That concludes our session on Cloud Foundry. In a couple

seconds we will tee off the next session.

>> Thank you.

>> CHRIS FERRIS: Welcome back, day 3 of Open Source Week. We are going to have an update on the Apache OpenWhisk project from Matt Rutkowski, senior technical staff member at IBM, and he was the original proposer for the OpenWhisk project at Apache. He is a current member of the PMC there and the committer on OpenWhisk. Take it away.

>> MATT RUTKOWSKI: Thanks, Chris. The OpenWhisk project is a serverless project, you can tell by the tag line on the title slide. If you attended the sessions already, you are familiar with the format. We are going to talk about the OpenWhisk community, how it was created, its history, about a technical overview of the project components, we will give you the status of what community is at and talk about its next steps and talk about how it's used at IBM, and we will tell you how to get involved.

What is Apache OpenWhisk? It is called serverless but we like to say it's more a function as a service platform, it's all about running functions as workloads as quickly as possible without the consumers or users being aware of any of the competing or any networking resources involved.

It looks to work against events, event driven. It is all about integrations with event sources, from data stores, message queues, mobile applications, Web applications, sensors, you name it. Anything you can conceive in nature can be turned into an event that a serverless function can analyze or handle or execute on.

Out of the box we support many functional languages. Node.js, Python, JavaScript and PHP we added within the last six months. In fact today we reviewed a independent contribution for go runtime, you will be seeing that go runtime classified be added shortly.

But we provide a general SDK, where you can bring your own language or bring your own runtime. People take runtime that we provide and build on top of them, add your own libraries. We have good documentation to do that. That is how the go runtime got contributed and got started as SDK. Now it is going to a native runtime service.

Highly performant, load balancing and work toward auto scaling, our goal is to have less than ten millisecond load times for your runtimes, function as quickly as possible. It's about being robust and composable. We have tooling I'll talk about, composer conductor. We had a model where other serverless platforms do not have a competent model based on functions. We want a model that people can extend and build upon and use to create robust work flow and applications from.

Every component, I need to point out, and I'll do repeatedly, is that all components are containerized using Docker. Everything you see in our architecture is intended to be packaged as a Docker container and public Docker hub. You can pull down all of our images. Our focus, right now we are moving to make sure we can be auto scaled using [inaudible] significant focus of our project going forward.

How to get started, IBM research. They had a notion of serverless computing with a strong model that could be used and adopted by developers. They started around three years ago. As soon as it was ready for public consumption, beta form, they put it out on public GitHub under the Apache 2 license, around February of 2016 and started demonstrating it at conferences. But they wanted to move it to a general governance group consortium, and we had at that time partners that were interested in that and helped us to take these source code so ended up taking it to the Apache Software Foundation, and were adopted as incubator project in late November, 2016. The Apache project is currently a set of 34 repositories, standing things such as core platform components, command line interface options, integration with different language runtime join, we provide different packages which we would love to grow and invite people to add to, out of the box, message keys, data sources which we have a CouchDB variant being worked on, push notifications, alarms, periodic trigger, RSS feeds that a group in our China research team contributed.

Choice of deployment, when we put it out there as open source, we started with straight VM deployment to containers, push for configuration but Red Hat became involved and pushed towards Kubernetes. We evolved towards that. Adobe is big on Apache Mesos which is a compatible framework. We have a small group of [inaudible] you have different choices for deployment if you choose to run this platform.

We have many repositories for tool integration, provide samples, performance tests and other things. Time line from top left to bottom right, IBM started the research, began coding in 2016. A lot of things I just described, Adobe basically joined officially in August 2016. We became a incubator on November 23 officially. General availability of the IBM version became available on BlueMix in December of 2016. It is known as IBM Cloud functions. We introduced integration I talked about such as message hub and Web actions, in February, March, those are some of the big integrations we add. We are adding more every month. We added a gateway. We were one of the few projects that has a integrated gateway so that is significant to make your actions available on the Internet as a whole. [inaudible] joined us in June 2017 and is a strong contributor to Kubernetes

[inaudible] interface I'll describe as well.

(audio is very muffled).

How do you engage, we have a project website which is [inaudible] best way to engage the community is GitHub. Of is [inaudible] through issues [inaudible] anybody can contribute [inaudible] make contribution [inaudible] Apache [inaudible] quick and simple. The team enjoys and monitors routinely various public Slack channels. That is the best place to ask questions and get involved. We have very open conversations about where things are going in the open source project as a whole. We have a Twitter feed where events [inaudible] other things like that. Apache events, we have a [inaudible] coming up in May which we are targeting for announcement. We had a blog site, if you want to see how-to and getting started from just the Apache OpenWhisk point of view that is a great source, place to start with. We have a OpenWhisk YouTube channel. I host [inaudible] zoom meeting and you can actually see the latest presentations, the proposals of projects, the go runtime was committed this morning as well as composer. [inaudible] YouTube channel [inaudible] always catch up and [inaudible]

We do design work there [inaudible] we have done some of our Kubernetes design work and some of the [inaudible] comments and things be captured.

Let's look at the high level architecture. I would describe it from left to right, everything in the green box I would call the OpenWhisk platform, which processes your functions from creation to [inaudible] on the left I show we have a CLI in go language, four different language runtime (background noise) integrate Python, Swift. Other languages, other languages you are interested in, we have a router, which is [inaudible] controller is the brain of the system [inaudible] activator [inaudible] that manages all of the entity, you would send the functions stored there, activation records and general state from [inaudible]

(sorry, audio is extremely muffled, I can't understand).

Looks through a whole sets of clusters, pulls container pools in Docker containers which in turn run different runtime Docker containers that run your function and communicates to those containers over Kafka. Kafka is also used for initialization configuration communication. We are trying to [inaudible] image all Dockerized, we are trying to make them all self-starting, self-healing, self-managed. We moved from a Agile base configuration deployment, inject configuration data, pull configuration, everything self starts and image communicates to Kafka through [inaudible] I'm enabled, I'm available, I'm down, I'm not functioning through [inaudible] it's a nice clean system, and we evolved to supporting [inaudible]

If you look at the model how we interact and think of events, how to process through the system. We start at the left, you have event sources which I described that can be anything, we can codify and put on a message of some kind. We have these things called feeds which can be provided as packages of actions that can manage these incoming events in the raw, convert them and make them ready for consumption in JSON format [inaudible] configuration control mechanism that feeds, so you can turn on, turn off these feeds, configure the feeds.

(voices in the background).

There are certain types of feed services. I won't get into that. Triggers are a [inaudible] channel, once the [inaudible] message the general event data, they can be passed to a main trigger, which the channel to an action, they are connected by a rule which may turn on and off. You can control programmatically, turn on and off the ability to send actions to the functions themselves, the rules. The action itself [inaudible] runtimes that can run your functions that I described earlier.

We also have various tooling options, we have a composer tool which helps you create the model. We have sequences which I'll show on the next slide in a diagram. We can do complex compositions for these things. Actually we checked last week something called conductor which can manage state of a large set of actions, we can run various [inaudible] switch statements [inaudible] processing, so you can actually compose these into another action, then the control will be able to manage the state and control flow to the actions and be able to determine which actions to load so everything is not static. It is dynamic in nature, and the control itself will be able to manage these compositions.

Simple view of [inaudible] manage the system, you can see three types of event sources that I show on the right, one is data store, that is a huge one. [inaudible] we talked about adding, we have couch and other data [inaudible] trying to add mongo right now. [inaudible] some cases we have a [inaudible] that works with them as part of a feed package. We can take those events and monitor and change their data. If you put documents in the data store, change documents, we can actually create actions, events out of them that go through actions. Social media is another source. Anything you do on the Web, upload a picture, Instagram, send a message to Slack channel, post things to Facebook, we have sample applications that show you how to do these things and actually sample catalog packages that offer these services.

But it can be low level things too [inaudible] devices [inaudible] instrument machine such as [inaudible] management

which you can actually track the health of vehicles and their status on the road and track where they are at, very powerful. You see from left to right the ingress of the messages into the system. You can see here that these actions are run and invoke the containers, single action [inaudible] sequences of action, different runtimes and languages and different actions and the second one from the top is Node.js, output to Python [inaudible] share which language they are written in the functions themselves, you can compose them in different ways. This is a straight line [inaudible] composer which are introduced into the controller in the last few weeks. All these complex programming idioms.

What is the status in the community? I grabbed this over the weekend. 44 repos, that is [inaudible] runtime repo this morning. We are having a conversation that Kubernetes is managed in a different repo [inaudible] we are talking about [inaudible] mainstream OpenWhisk platform. That repo might go away. Plus one, minus 1. We have 2836 stars, that is the repo alone [inaudible] ten to 20 per week and momentum, the number has gone up. My personal gauge of health, it's the number of people contributing and creating forks. That means they have taken the time to, they want to net change the source code and look at it and run it in their own repo. We have I believe over 208 active contributors, contributors who have merged code. That does not include people who have proposed ideas or discuss [inaudible] contributes to the main repo.

Versioned releases forthcoming I'll show in next steps.

What's next, we want to grow a community. The big thing to help us is getting towards graduation. We are incubator project in Apache. There is a criteria, we must pass certain milestones and show the community is growing and part of that is showing that you have a release process, all the legal requirements. We demonstrated this morning that we can now identify 12 repos as core repos. Our first step is [inaudible] source code, I have a repo created where we have a manual process described and automated process [inaudible] as release manager we can package up all the source code for all 12 repos, send it to a staging [inaudible] Apache location along with the correct [inaudible] we need to formalize that more. Maybe in the next month we will have releases on a regular basis [inaudible]

>> You want to clear the calendar.

>> MATT RUTKOWSKI: Sorry. Extensibility is huge [inaudible] our partners want to extend the system. They don't want to use CouchDB for artifact storage. They don't want to use Kafka. They want to use [inaudible] this idea has caught on. It has helped IBM itself in moving forwards towards Kubernetes, in that the load balancer is credible. We have moved from a load

balancer which is pretty round-robin-ish to one that is doing charting. We have been able to test it out because we created a interface. It is proposed, we created micro, we broke up parts from the artifact store to separate logging, log provider, tracing and debugging provider. Having is being thought of in terms of how can we make it [inaudible] fingerprint of what makes a interface and architecture stickable and viable and encourage people to code to it. That idea is taking off.

In terms of frameworks, I mentioned pure config that is still ongoing. My personal goal and our goal is basically go to zero ansible, avoid push configurations and do pull configurations. In fact we had a great milestone this week, in that our load balancer used to be binary, the main actor in a stand by when we [inaudible] now we are fully scalable, number of controllers, each of them able to work together with shared state. This is a great story. It's a great open source project story.

My personal [inaudible] improve integration, we never have enough integration. Service computing by itself means nothing. It has to work against events that come in that need integrations with event providers, feed services and we need more tooling to be able to engage people through their I.D.s or command lines or whatever. It's about needing to reach developer [inaudible] we are going to be creating a Web management [inaudible] month's time, we have somebody working on repo [inaudible] hopefully it will be ready for prime time in a month or two.

Plugable API, 7 different language for [inaudible] we have to add commands [inaudible] one of the big things, what I call multipliers, general integration, anything that handles events in messenger, integration source so MQTT package that reached development that never made it to the community, open source community at Apache. Rabbit MQ has been asked for by popular demand, again we need people to come to the project and help. That is a nice way to how do you get engaged and get started.

At developer [inaudible] language and start using it. You can talk to us about expanding our runtimes. That is a routine conversation we have nowadays on our Slack Channel. I'm trying to use this Node.js thing [inaudible] library, what things we package in runtime [inaudible] negotiable [inaudible] providers, we want hub with integrations, we have samples out there. We will help you. We have some application or some service, we will help you do that. We have good [inaudible] you can start with one of our sample applications, if you want to check out patterns [inaudible] help you with that. We have generally architecture patterns. Anywhere you are coming from, any application type we have a sample for you to start with.

In terms of if you want to become [inaudible] in the project

itself, Apache [inaudible] we have a list you can subscribe to. Please join that. Developers themselves love Slack although it's not officially recognized as a communication channel for any decisions at Apache, we definitely engage there and if the conversation becomes one, let's start proposing this, we will take it to the [inaudible] I get this error message, whatever, it's [inaudible] of course we have code patterns [inaudible] some part of OpenWhisk, use the IBM incarnation of it, if you go next month to the Adobe conference [inaudible] look for any conference, people in serverless [inaudible]

If you are a developer, our skill set is [inaudible] as far as client development, go lang is huge, all the things [inaudible]

(completely indecipherable).

Python tooling, native [inaudible] we appreciate any help with framework. When you have a Docker, Kubernetes, Mesos we never have enough experts in that. I know that we will have some releases of Kubernetes, you can [inaudible] I'm sure we are going to have to have people making multiple [inaudible] variants for different components [inaudible] different type of options to different [inaudible]

If you see an issue, documentation is always good on any project. But we try to on the [inaudible] good first issue, look for those. Again ask on Slack, I'm interested in this, can I help out. And guaranteed, you will be directed to some place where you might be able to help fix an issue.

Call to action, get involved. Meetups, all the standard things. Please subscribe to IBM code. My personal day-to-day job at IBM is to try and get more patterns, more things in the serverless portion of IBM code. You will see a lot of activity. We have 13 patterns lined up for IBM code in 2018. We have three at large patterns being developed now. There is a lot of activity, patterns on IBM code.

That is all I have. Questions, is that realistic.

>> CHRIS FERRIS: Thanks, Matt. I don't see any questions. I don't know if anybody else has been monitoring the chat. If not, we can say thanks to Matt and proceed with the next session. Thanks, everyone. Hopefully we have Susan on.

>> Yes, I'm on, can you hear me? I can share if you would like me to.

>> CHRIS FERRIS: Please.

>> SUSAN MALAIKA: Is Raj on? Raj, can you speak, can you hear us, can someone ping him? (chuckles).

>> Raj here.

>> SUSAN MALAIKA: Great. Excellent. Can you see my slides? (door closing).

>> Yes (overlapping speakers) (background noise).

>> Presentation mode.  
>> What is that?  
>> There you go. (overlapping speakers).  
>> SUSAN MALAIKA: Yes. I'm going to watch the chat as well,  
I think.

(whistling in the background).

>> CHRIS FERRIS: Welcome back, everyone, day 3, session 3, Open Source Week here at IBM. Next up is our session on the open data platform initiative or ODPI. With us today we have Susan Malaika, senior technical staff member in IBM in the open technology group and Raj Desai. Take it off.

>> SUSAN MALAIKA: Okay. Welcome, everybody.

(voices in the background).

This talk is about the ODPI, as the ODPI is a consortium which was started in 2015. My colleague Raj will also give part of this presentation.

So moving on, let's go to the next slide. Can I go to the next slide? Yes, I can. Today's agenda is, what ODPI is, how it was created, where the community is, and then a small technical overview, and the current status and what's next, and then IBM and how to engage.

I will say that this, giving this talk today is probably rather at a odd time because we are in kind of an in-between stage at the ODPI, with some work sort of getting into a steady state, and some work launching. But keep that in mind as we go through this presentation that we are in this sort of in between stage. But it does mean there is a lot of opportunities for people to join us, because we do need to ramp up on new activities.

The ODPI is a nonprofit organization. It started, as I mentioned, in September 2015, and its goal is to simplify and standardize the big data ecosystem and increasingly any data ecosystem to make it easier for enterprises to mix and match their tools, and their software. Towards the end of last year, the ODPI determined that it will increase its focus and attention to the data governance activities. So a new project management committee was launched led by Mandy Chessell, a IBM distinguished engineer. This new PMC was launched with Horton works ING group and ODPI members. You might find it strange that we have maybe a company ING which may not be a typically technology company as part of this effort. But you will see why in a moment. We will expect to see more diverse industries joining the ODPI.

The way the ODPI is run is that there is a Technical Steering Committee, and a number of project management committees, and in fact, we had our steering committee meeting this morning which was great.

And you can find the charter at the link on this slide. It was created in September 2015, a number of companies got together to solve the challenges for big data companies that are integrating across different platforms. They might be buying software from different companies and were finding it difficult for their applications to run without change across the diverse data platforms.

The ODPI also runs a user advisory council to do its work, so the customers on the council may give suggestions about how certain activities should proceed or what their needs are, and how we can better satisfy them through the ODPI.

The MPI delivers webinars and education, and participates in conferences and we will list those in a moment and also participates in meetups.

The data governance activity in particular which was recently launched is, will make it easier for data science and machine learning projects to proceed because it will provide a unified approach for handling metadata and governance, making it easier to discover data for analysis. It will also help enterprises apply directives through the various compliance initiatives they have to adhere to these days.

Moving on, so the community operates on GitHub and there is an ODPI GitHub. And as well as producing specifications and tests, the team at the ODPI, the people who participate in the ODPI very much engage with Apache projects as well. Some of the Apache projects that the ODPI members often engage with include atlas, hadoop, hive, big top and also Jupyter Notebooks. There is a special-interest group at the ODPI that is looking at notebooks and data science as we speak.

(clicking noise).

Now some of the members, their photos are on this slide. Mandy we already mentioned. Raj, who is about to speak about the work that he has been doing and leading at the ODPI and Alan Gates, John, Madhan, Ferd and I wonder if they are in the chat, I will look in a moment but I'll hand over to you now, Raj and I'll share the GitHub address about the ODPI runtime PMC so that you can tell us about that. I'm just moving over. This is the link. Can you see it, Raj?

>> Yes, I can.

>> SUSAN MALAIKA: Am I in the right place?

>> Yeah, that is fine.

>> SUSAN MALAIKA: Please go ahead.

>> All right. Yes, so I've been mainly working on the ODPI runtime. So to describe what ODPI runtime is, I'll go through two main deliverables that the PMC produces. One is runtime specification and the other is runtime compliance.

First, what Susan is showing on the screen is the runtime

version 2 specification. The runtime spec is essentially compatibility guidelines for applications running on hadoop platforms. The goal that we look to achieve by having this specification really affect three main groups of people, one the end users, second is the software vendors and also the platform vendors.

For end users, it provides the ability to, since you run any application on any hadoop platform and have it work, I mean that is kind of our main goal, is kind of the ease of use of applications running on hadoop, essentially as easiest way to do it. For software vendors, the compatibility guidelines enable them to ensure that the applications are interoperable across any platform. Lastly, for platform vendors, the compatibility guidelines enable applications to successfully run on their platform.

So that is kind of the runtime spec as a snapshot. I guess in detail, the spec covers core components of the hadoop ecosystem. This include Apache, hadoop, V27, yarn, map reduce, hive and HDFS.

So going on to new versions of the runtime spec, we foresee to support different versions that are coming out. The second main part on that kind of ties in directly with this runtime specification is the runtime compliance. This is actually built on Apache big top. Any platform that can essentially claim ODPI runtime certification by downloading and running our acceptance test suite, and run it against hadoop cluster, so if the test passes, a number of the ODPI PFC will review it and merge it with our compliance repository. As well, a number of the ODPI, marketing team will follow up with trademark assets and guidelines for advertising ODPI compliance.

Also, two other certifications we offer are the operations compliance and being ODPI interoperable.

Some of the next steps that the runtime PMC is looking to do ask make enhancements on the current runtime acceptance test suite, making it more informative than kind of binary pastel. We are also starting to develop the natural release of ODPI runtime version 3 and the acceptance test suite. So this will involve a spec around the HTP version 3 distribution and test suites running the new features.

If you are interested in joining runtime PMC, feel free to E-mail me at my E-mail address, that Susan linked it in the title slide. That's pretty much it.

>> SUSAN MALAIKA: Thank you very much, Raj. I see John Mertic is in the chat. He is another person you can ask if you want to join the ODPI, any of the PMCs.

So that is John there in this picture here. I see him on the call. Okay.

Now let's continue. So what Raj described to us is work that's been under way since the inception of the ODPI, to define compliance tests with hadoop and Hive and so on, projects that come from Apache, where you want to be sure that your distribution or your version that you deliver of these Apache projects have consistent interfaces, so that enterprises installing those projects, those distributions, will be able to run their applications on the different versions, on the software that comes from different sources.

That work has been under way and still continues.

So, the next slide, set of slides, are describing a new PMC, a new activity at the ODPI around metadata and open governance. That is the activity that I mentioned as being led by Mandy. Metadata is stored in companies in all sorts of places. But increasingly, people are using repositories, so that is at the bottom of the layer at the bottom of the slide that I'm showing.

But these metadata repositories offer different interfaces, and different ways of exposing the metadata. So an activity that is under way that's taking place in the Apache Atlas project is to define common interfaces. So open metadata repository service interface is OMRS, and then interfaces is called open metadata access service OMAS. OMRS is databases repositories that comply would expose, and then tools that need to access repositories or need or put data or modify data, metadata repositories, can use the OMAS interfaces, and can offer the OMAS interfaces to access the repositories.

Apache Atlas itself is a metadata repository and is kind of the work under way to support these interfaces is kind of to make Apache Atlas a reference into implementation. But then other metadata repositories can also offer these same interfaces that are being defined and implemented in the Apache Atlas projects.

When these tools and repositories that expose and support these interfaces will then be able to exchange metadata and to store it in diverse repositories and different tools will be able to be used on various repositories, so that's a key element of what this open metadata initiative is about, to be able to mix and match metadata across repositories and across tools within an enterprise.

The other angle to this work is to make it easier to adhere to the various compliance initiatives that are coming out, and one that is listed at the top on the right side of the slide is the GDPR regulations in the European Union and any country or any entity that deals with data in the European Union has to support this and be able to show that it supports this regulation. By having this common way of dealing with metadata in place, it will, we expect it to become easier for companies

to be able to show compliance with these regulations, and indeed, be able to keep up when they change as well.

Also listed on the right-hand side are other standards that we will be looking at, because the work that is going to take place in the ODPI and that is beginning to take place is about making it easier to apply these regulations. So the work in Apache Atlas project is about coding the interfaces, and then the work in the ODPI is about making it easier to apply compliance regulations.

So open metadata and governance is made up of these, a way, an agreed way of describing the data types, metadata types, agreed way of doing the APIs interchange between repositories that contain metadata and the various tools that will, and software that may be accessing metadata not as its primary goal, but has to access metadata as in order to achieve its goal. Then the frameworks that are automated governance, so this is the scope of the work. And by having the tools in place that support these agreed upon interfaces and meta data types, you will be able to have catalogs in your company that will list all of your data, where it's located, its lineage or provenance, the structure in it and in an agreed upon way, and new tools that adhere to these interfaces will be able to integrate with your data catalog out of the box.

As new data comes in to your systems, if you use tools that comply, then your catalogs will be automatically updated and with new metadata that's about the data that is coming in. You will also need subject matter experts like experts in GDPR, experts in privacy, and so on, to be able to describe the rules that need to be applied to all this metadata. And indeed, the new PMC that is being set up for the data governance at the ODPI is going to flip-flop each week, so one week we will have subject matter expert type discussions, and then the next week we will do metadata technical discussions.

So because it's very critical to have subject matter experts engaged, and it will be, easier to automate the processes. The ODPI is going to work on producing packs, value add packs by governance or regulation ruling, and the value add pack will have guidelines, it will have best practices, it will have model extensions for in support of a particular domain, a particular regulation. And it will have glossaries and rules written in a JSON notation so that it can be, these rules and regulations can be processed automatically by software.

These, there will be plug-ins to, in support of these individual value add packs. For example, a value add pack for GDPR. So that's the work of the, that is happening in the data governance group at PMC at the ODPI.

You can see a link there at the bottom, where you can see the

related work in the Apache Atlas project. What is the current status? ODPI launched this PMC and the work that Raj told us about continues, and we still need people involved in the hadoop compliance and related software such as Hive and so on. The PMC group is recruiting. We need data governance metadata experts, and John who is on this call and in the chat, you can contact him in the chat, you can do that now, and his E-mail is also on the slide. You can connect with him if you want to join this group.

Also we are interested in having new members at the ODPI, new clients who could perhaps be on the board, or in the custom advisory council who want to shape the direction, like which packs, which governance areas should we work on first. There are a number of related ecosystems that we are connecting with, like the metadata and governance communities. There is something called gamma in the U.S. that is a metadata community that has been around a while, that is just an example.

What's next for the ODPI? So stage 1 is happening as we speak. One of the first things to do is to make a value add package concept and gather feedback. The early stages have happened, and now we are into the next stage of that. Eventually we will get to the stage where we complete based on the feedback we obtain from the PMC and any subject matter experts. The ODPI at IBM, that was very active in the Apache Atlas project and in the other projects that Raj talked about earlier, hosting meetups. We have some data governance sessions at Las Vegas, if you just go to the link and look, some of us will be there. Mandy, Shrikant, John and myself a number of us will be running, there is a particular panel we have on this topic. We are looking to identify clients to join the ODPI advisory council.

I just listed a number of products, solutions, just to give you an idea of the kind of software that may be interested in this area, so security solutions, data platforms and so on. These are the kind of, that catalogs, these are the kind of software tools that will gain a lot from supporting and participating in the governance work.

And to shape the world of data governance through the ODPI, participate in the PMC, join the mailing list, the Slack channel. We are present at the data works summits and the strata conferences. The strata conference next week in San Jose, some people will be there. I think Marina, I don't know if she is in the chat, Marina from ING will be there. The data works, the strata conference next week, the data works summits in Berlin, coming up. Mandy and John and a number of other people will be there. Also there is a great data governance webinar that's been delivered and we are working on a couple

more governance webinars. It's a far-reaching topic.

It's a new topic, and really rather exciting, because this is brand-new work in this area of data governance, trying to automate it more. And then the call to action, essentially join the communities, and start contributing whatever way you can.

We don't just need programs, we need subject matter experts as well. We haven't done any IBM code patterns yet, but there will be further opportunities in code patterns, tech talks around this area.

That is the end of my talk. Are there any questions? Or have there been anything happening in the chat?

>> CHRIS FERRIS: Thanks, Susan. This is Chris. I've got a question. So obviously, the work that you are doing in ODPI, you and Raj, essentially reads on projects elsewhere, right? So Apache hive, hadoop, Spark, so forth. Can you describe maybe how the communities sort of (overlapping speakers) align their efforts, and what the process is for, you know, ensuring that everybody is sort of moving in the same direction.

>> SUSAN MALAIKA: Yes, absolutely. Very often, the way that happens is by having the same people engaged. Typically, we will have the same people who are active in Hive or hadoop also be part of the PMC as well, so that we can keep aligned.

What happens, the most of the programming goes on in the Apache community, the basic programming, for the projects themselves.

>> CHRIS FERRIS: Right.

>> SUSAN MALAIKA: What happens in the ODPI, is a software and guidelines and tools that sit on top of the software in Apache, and may cross boundary, you know, they may relate to three or more projects, they could be compliance tests, they could be these packs that I talked about, that will implement GDPR on top of Apache projects, for example. So but we keep things in sync by having the same people.

In fact, we actually went back and forth on the Apache Atlas and governance work. We did think we would do some of the programming work for the interfaces in ODPI. But then we decided it was too complicated for keeping the deliveries in sync across ODPI and Apache.

>> CHRIS FERRIS: It's a very collaborative effort then, I gather.

>> SUSAN MALAIKA: Absolutely, absolutely. As I said the main people are, I don't know, Raj, if you want to say anything with your experience on the compliance test and so on. I'm not sure if he is still on.

>> Raj: Yeah, I mean just to I guess hitch on to what you said, the more people that participate, the better standards and guidelines that we produce. From all areas, the platform, to

vendors, and anyone that is participating in the ecosystem, because sometimes a person can have a different point of view, depending on what their software does, so very much a collaborative effort and (overlapping speakers).

>> SUSAN MALAIKA: To latch on to what Raj just said, you know, the people in the Apache projects tend to be the programmers, developers, we are trying to reach out to subject matter experts in areas beyond the project itself, such as the governance regulations in different, like a privacy expert or a GDPR expert and so on.

>> CHRIS FERRIS: Awesome.

>> SUSAN MALAIKA: So collaboration with the Apache projects, having the same people who are in the Apache projects join us in the ODPI meetings and bringing in more people from other areas in the ODPI.

>> CHRIS FERRIS: Great. Okay. So I don't see any other questions in the chat. So (overlapping speakers).

>> SUSAN MALAIKA: I can't believe it.

>> CHRIS FERRIS: I know (chuckles) it's a fascinating topic. It sounds like also a very important one. Thanks again, both of you. That wraps up our third session today on day 3 of Open Source Week. We will continue in a moment with Tensorflow. Thank you.

>> SUSAN MALAIKA: Thank you.

>> CHRIS FERRIS: Tom, are you on?

>> TOM TRUONG: Hi, Chris, yes, this is Tom. I'm on.

>> CHRIS FERRIS: Want to switch ... can you share your slides?

>> TOM TRUONG: Can you see the slide?

>> CHRIS FERRIS: Looks good. Is Jim on as well?

>> TOM TRUONG: No, unfortunately Jim is (overlapping speakers) I'm flying solo.

>> CHRIS FERRIS: Okay. Let's get this party started. Welcome back, everyone, to the fourth installment today on day 3 of Open Source Week here from the open source team at IBM. I'm Chris Ferris, CTO for open technology, and we are going to be moving on to a fourth presentation today, which is on Tensorflow. It's going to be presented by Tom Truong, Tom is leading a team of open source developers in the Silicon Valley lab of IBM, and his focus is on data and AI open technologies. He's been working with Google and the Tensorflow community, working to bring that interest, to get IBM's involvement up, and to work more closely with the Googlers. Tom, you want to take it from here?

>> TOM TRUONG: All right. Thank you, Chris for the introduction. Hello, everyone. My name is Tom Truong, program director of the cognitive technology and system performance

group at IBM Silicon Valley lab. I lead a team of engineers, subject matter experts working on code development and performance project to focus on open AI technologies.

(muffled audio).

For the next 25 minutes or so I will give you a brief overview of Tensorflow which is a framework for deep learning from Google and how IBM is contributing to this community. The presentation is meant to be preview to whet your appetite. I hope you will dig deeper to learn more about this exciting open source technology. For today's talk I would like to take you through the following topics, what Tensorflow is, how the technology was created and where the community is. I will give you a brief technical overview on the current status of Tensorflow including activities inside IBM as well as outside in the community.

I will end the presentation with let's get started and call to action. Given the time constraint, I will not be able to go into lots of details. There are many hyperlinks provided throughout the charts so that you can learn more about Tensorflow.

What is Tensorflow? It is an open source software library for deep learning. Initially Tensorflow was developed by researchers and engineers working in the Google team within Google's machine and research organization. It was for the purpose of conducting machine learning and deep neural networks research. The system turned out to be generally applicable in a wide variety of other venues as well, afterwards it was under Apache 2.0 software license released. It is preferred open source tool for AI and GL nowadays and used extensively by using internally.

You will see there are growing numbers of contributors from other organizations and companies as well. I heard analogy made recently at a presentation by Google. The goal is you should be able to build an AI application as easy as it is for you to jump into a car and drive it, without knowing much about the car mechanical and electrical systems. That is the analogy.

As you may see the software platform system nowadays provide building blocks like Lego to be high level applications. The diagram on the right show the high level general architecture of Tensorflow. The runtime library is basically a cross platform library, and there is a CAPI layer that is separate the user level code, in the multiple different languages from the core runtime, and the core runtime is shown here in the larger box that encapsulate the lower devices, the networking layers, the kernel and data flow component.

Tensorflow supports multiple languages but using has prioritized Python and C++, mainly because the users are most

familiar with these languages. As features become more [inaudible] they port them to C++, so that user can access the optimized implementation from other languages. There are also many open data sets available for people to use. Since deep learning requires massive amount of data to build recognition models, pattern recognition models.

This is more about how Tensorflow was created. It was a second generation deep learning system, successor to Google dist belief. The famous Google team built disbelief as a machine learning system and using neural networks back in 2011 and Tensorflow was derived from it, to be a faster more robust deep learning system.

[inaudible] long history going back to [inaudible] from NYU and these researchers [inaudible] only until 2012 where we had machine fast enough, we have [inaudible] DL began to surpass other learning methods. Jeff Dean and [inaudible] let the release of Tensorflow from Google. It becomes the unifying system used for all internal Google projects. For open source activities, however [inaudible] Google technology. Googles [inaudible]

(muffled audio).

AI. You may know general [inaudible] torch and others, open source, they do have [inaudible] but Google Tensorflow is still the most dominant.

Where is the Tensorflow community? The main [inaudible] have been Google employees but there is a growing number of contributors from other organizations and companies, including IBM, as shown in the picture on the upper right. Jeff Dean from Google shared some interesting stats recently which demonstrate that you take, that Tensorflow is a vibrant and popular community, there were more than 475 nonGoogle contributors to Tensorflow version 1. They completed more than 15,000 commits in 15 months. That is about 1,000 commits per month. That is very impressive. And [inaudible] like 80 issues that users submitted in GitHub. The graph on the lower right shows the top contributors with long term engagement over six months.

Here we show the links to the GitHub repository, the official website and Facebook pages. There are lots of useful information to get you started. The community is very very active.

It's interesting to note that approximately 7,000 GitHub repository include the word Tensorflow in the title last time I checked. It's very popular.

Here I tried to give you a brief technical overview. But the technical website is best, is the best source, that is there with excellent resources, including documentation, performance benchmark and news about conferences, etcetera. The main

Tensorflow Web Page is maintained by Google. Google announced that release 1.5 now is the latest release, with the most stable release usually one version back so that is 1.4.

The background shows Tensorflow execution engine, can be run on top of various CPUs, GPU, Android and [inaudible] devices. Google prioritizes Python and C++ front end, with Python being the most supported. Core TF components are C++ for speed. The background shows Keras, it is a Python deep learning library with high level neural networks API. It is capable of running on top of Tensorflow, CMT [inaudible] it was developed with a focus on [inaudible] experimentation. It [inaudible] Tensorflow for the mass so it's quite pop layer. It has become the first high level library added to the core Tensorflow at Google so we would expect to see that it may become the Tensorflow default high level API in the future. There are other high level Python neural network libraries that can be used on top of Tensorflow, such as TS slim and others but these are less developed and not part of the core Tensorflow.

I also found out that ecosystem available in GitHub include example how to integrate Tensorflow with other open source framework, including Docker, Kubernetes, hadoop and Spark. The example [inaudible] template that a user can tailor for their use cases and there are valuable resources for developers. I encourage you to check it out.

What is new with Tensorflow 1.5? Obviously there are changes and [inaudible] latest release [inaudible] there are three major features in 1.5. The first is eager execution, which enables faster debugging with immediate runtime errors and integration with Python tools. According to the GitHub read me file, it is a experimental interface to Tensorflow and that provides a imperative programming style and [inaudible] when you enable eager execution, Tensorflow [inaudible] immediately and [inaudible] easy to use Python control flow. That is [inaudible] feature in 1.5. Second is Tensorflow lite which supports [inaudible] with low latency and with a small binary [inaudible] mobile applications.

Let's say you rebuild a model to classify a image, now using Tensorflow lite that same trained model can be deployed to the mobile device and then you can use it and the application can classify the image directly from the device.

Thirdly [inaudible] CUDA network library [inaudible] now built in.

What is the current status of Tensorflow, version 1.5, it is easy to find out status by GitHub repository. You will find files [inaudible] find out status of builds and targets. The latest snapshot, Tensorflow in GitHub shows [inaudible] definitely an active community compared to others. The graph on

the left, shows a recent snapshot showing the examples, it shows how active TF is compared to others. There is a huge activity for Tensorflow around 2016. [inaudible] of the GitHub.

Moving on, what is next for the Tensorflow community? As you may imagine Tensorflow community is growing like wildfires in just over two years. They got a lot of help from the alpha go victory, it has risen to dominance in deep learning. But also, as noted earlier, it's Google's application and its website and performance tuning work has been the key reason why it has become so popular.

Also more than 500 active contributors submitted code with 1,000 commits per month. We found out that major universities have started to teach machine learning class using Tensorflow, thus adding to its popularity.

Google hosted the first DEV summit last year at its headquarters in mountain view, with live streaming video to communities all around the world. It is shown on the map on the right side. This year, the second annual Tensorflow DEV summit will be held on March 30 at the computer and history museum in mountain view. Like last year there will be live streaming video to communities around the world.

But again due to limited space and extreme popularity, you will have to apply on-line, and only after you are invited will you be able to attend the summit in person. For many we will have to form a watch party and participate in the summit remotely.

Also related to Tensorflow is Tensorflow processing unit. Last May Google announced a second generation TPU available on the Google Cloud [inaudible] for both training and [inaudible] on IBM side, we see a lot of activities around Tensorflow. Many IBM products and solutions offer Tensorflow framework as part of their AI package, for example, Watson ML and more on this on the next chart.

I show here the [inaudible] support Tensorflow. IBM research is increasingly using Tensorflow for research project as well. Customers who don't have their own power system can access the new software of [inaudible] my team has access to a environment and we have been using it for evaluation of various AI workarounds. That is how we do it.

At IBM code website for the wrappers you will find Tensorflow related patterns, and the number of patterns is growing. My team has been given hands on programming to toy around in Tensorflow, both inside IBM and outside conferences, like at the [inaudible] San Francisco or Seattle. Next month we will do the same in Helsinki Finland. [inaudible] train it and use it [inaudible] for example, to build a mobile app using Tensorflow that will recognize and translate handwritten Korean characters

[inaudible] images from the metropolitan museum of art. We do share some of that at the IBM code website. So check it out.

Currently, my team and I are part of the new Tensorflow distribute community. The focus of the community is to address issues that are related to Tensorflow build and packaging. We want to enhance TF mechanism, which is currently based on basal and we want to explore the use of C as a alternative. We are following the new [inaudible]

(audio breaking up).

To enable them to run on Kubernetes containers.

Let me start to wrap up. Where should we go from here? This is many links to websites, blogs and code pattern, classes and references on-line. Tensorflow.org is a great place to start. I like going there. We share [inaudible] IBM code website [inaudible] there are also many classes offered on-line as well. When you do a search, you will find easily hundreds of books on Tensorflow, for example SF iBooks on-line so please go out there and explore.

Here we are at our last chart, with a call to action. Again I hope you find this interesting and get involved. You can do so by attending the community meetup to learn and share with other developers. Participate in various conferences, subscribing to the IBM code newsletter and attending the code tech talks. Eventually we hope you contribute code to GitHub as well. For your convenience I have a link to GitHub in the meetup, the summit and the communities.

That is the end of my presentation. I hope you find this interesting and [inaudible] in Tensorflow, thank you for watching and attending Open Source Week at IBM.

>> CHRIS FERRIS: Thanks, Tom. Great. As he said, that concludes day 3 of Open Source Week here from the open source community at IBM.

Please tune in tomorrow for day 4.

>> MARC-ARTHUR PIERRE LOUIS: Chris, there is one question that is there in the chat room. Does Tensorflow support nonNvidia JPUs.

>> Say again? I'm looking at the chat.

>> MARC-ARTHUR PIERRE LOUIS: Does Tensorflow support nonNvidia GPUs.

>> TOM TRUONG: I believe so. It supports a variety of different GPU, including on the power system as well.

>> MARC-ARTHUR PIERRE LOUIS: Thanks.

>> From the YouTube chat, [inaudible] Tensorflow compared to other DL frameworks like torch and caffe.

>> TOM TRUONG: Popularity here you get a lot of support, it has a very active community. So if there are issues, things that you need addressed, chances are someone runs into a issue,

they are working to help you. There is a huge community effort. So that is a big advantage compared to others. The other ones have initiatives as well. But like I said Tensorflow seems to be the most dominant and most popular right now.

(dog barking).

>> That is questions on YouTube.

(dog barking).

>> TOM TRUONG: Thank you.

>> CHRIS FERRIS: Thanks again, everyone. Please tune in tomorrow for some other open source [inaudible]

(dog barking).

Take care.

>> Good-bye.

(end of call at 1:51 p.m. CST)