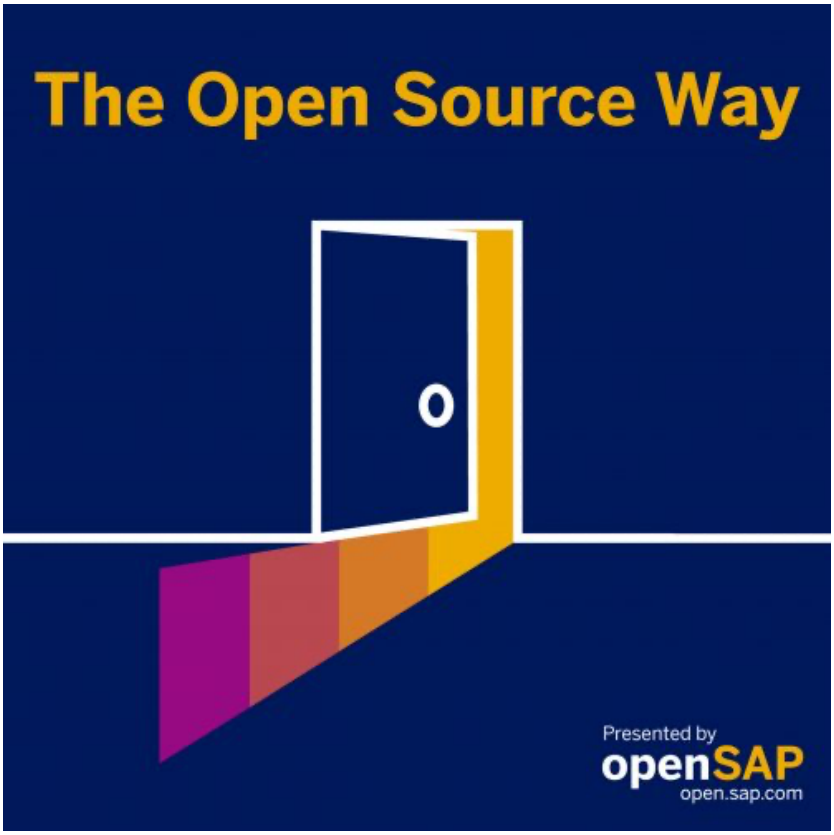
The Open Source Way

Episode 05 – Project “Gardener”– Managing Kubernetes at Scale



SAP Open Source

SAP

Karsten Hohage: Welcome to The Open Source Way. This is SAP's podcast series in which we'll talk about the difference that open source can make. In each episode we'll talk to a different expert and we'll talk to them about why they do it the open source way. I'm your host Karsten Hohage and this episode, I'm going to talk to two people at once. Let's see how this turns out. We'll welcome Vasu Chandrasekhara and Tim Usner, who will tell us about project “Gardener”. Tim started as a business software developer and changed to the Gardener project in 2018. He is passionate about open source. He loves to see the project evolve every day thanks to a growing and contributing community. Vasu started his first job at SAP as a technology consultant. He initiated the Gardener project and is in all ways and at all occasions an advocate for open source within the company. So Tim, Vasu...usually we only have one guest. Welcome. Why do we have the two of you?

Vasu Chandrasekhara: Maybe I'll take that one, Vasu. Essentially, Karsten, I thought it would be nice not only to hear from my perspective. I'm now more of an evangelizer of a business development role in the context of project “Gardener”. But I wanted to have somebody who is from the machine room who really can tell about how the discussion happens and pull requests on GitHub and so on. That's why I invited Tim.

Karsten Hohage: OK, great. Then let's say hi to Tim and maybe start off Tim with the first obvious question. We hope, of course, with the guy from the machine room, we will dig a lot deeper than in some of the previous episodes of this podcast here where it was a lot about open source processes. Tim, let's start with a simple one: The What-is-Question. What is Gardner?

Tim Usner: Hey Karsten, thanks for having me. What is Gardener? Gardener is a Kubernetes cluster-as-a-service solution and with it, end users get fully managed Kubernetes clusters on various infrastructures. The whole project is available and developed in the open source space and at SAP. Gardener-managed clusters are used to run all sorts of containerized workload. For example, business platform services.

Karsten Hohage: So, we are basically abstracting between the business platform by SAP and/or somebody else and a Kubernetes infrastructure, is that right Tim?

Tim Usner: Not just an abstraction, it's more an enablement of Kubernetes-based environments at SAP.

Karsten Hohage: OK, maybe we will understand even better in the process of the podcast. In the first place though after a rough introduction of what is Kubernetes, now, SAP usually builds business software. Vasu, how did SAP get involved with something like Gardener?

Vasu Chandrasekhara: Well, I think I have to give a little bit of a quick history and some background, you know, sometime around 2013, something interesting happened. Obviously, SAP was also noticing and also was experimenting with something that came up around the containers, with Docker specifically. And when you look at Docker back then that usage of containers was still mentally somewhat disconnected from a topic which we call the distributed systems domain, systems like Hadoop and Mesos. And what has changed around this mindset was when orchestrators around containers came about like Docker, Swarm, Nomad and especially when Kubernetes was launched. And when Google actually launched Kubernetes in a neutral foundation, called the Cloud Native Computing Foundation, under the auspices of the Linux Foundation, it became very clear that there will be a huge pull for this technology internally and externally, for Cloud Native environments and all kinds of tooling from this Cloud Native ecosystem. That's why we essentially knew that Kubernetes would be a big thing also for business software and that's when we entered.

Karsten Hohage: OK, so Kubernetes was obvious to become the distributed cluster approach, but which problems did we need to solve that brought Gardener about?

Tim Usner: OK, let me take this one and let me maybe begin with the start of our journey. So in 2017, a small team at SAP started to experiment with all Kubernetes and as you already said, by that time it already had promising features and gradually turned out to be the de facto standard to manage containerized workload. And what that team also noticed was that deploying and managing Kubernetes clusters is a tough and sophisticated challenge. So this is where also a lot of evaluation of already existing cluster managers happened. And of course, we also had offerings from the commercial vendors at our disposal. But they did not cover many of our crucial and immediate requirements, so we eventually opted for building it ourselves. And I can tell you that

was not a light decision, because right from the start we knew that we would need to cover thousands of clusters across the globe on all kinds of infrastructures. And the service, as well as the end user clusters, must scale and we would need to automate 100% of all aspects, because otherwise, as you can imagine, our TCO would still explode. So with that, our endeavour grew into launching the project “Gardener”. And its mission statement, in a nutshell, is a universal Kubernetes at scale which is at the same time a quite bold statement, isn't it? So we also had a nifty little secret that helped us along the way and we can openly lift that secret here. Project “Gardener” was not only built to create Kubernetes at scale, but it was also built in Kubernetes.

Karsten Hohage: OK, and being built in Kubernetes means what in the end?

Tim Usner: It means that we use Kubernetes to manage Kubernetes clusters which is also something often referred to as Kubeception. So Gardener is one of the early adopters of this concept called Kubeception.

Karsten Hohage: OK, and with that approach, in the end, like all that Gardener set out to solve, would you say it's doing that, maybe Vasu? Are we answering to all the market and technical challenges that we perceived?

Vasu Chandrasekhara: Well, internally at SAP, we're using Gardener to create work areas and infrastructure on various different providers and business software, which is written net new on Kubernetes is making use of Gardener. And so we have validated it internally and we are also exposing that as an open source project. So that's one of the nice things because here we have the ability to actually get into a conversation with our large customers and with the open source community to do co-innovation, to see what other requirements other people have with respect to managing Kubernetes at scale.

Karsten Hohage: You said we can get into a dialogue with our large customers here. So something like Kubernetes is mostly targeted at large organizations, is that right?

Vasu Chandrasekhara: Not only large organization, but in respect to if you look at what Gardener does. It's about management at the scale of these infrastructure components. If you only have one or two clusters and if you are in an environment that is quite homogeneous from start then I think you wouldn't need Gardener per se. But the

moment you have three clusters and you have them across various infrastructures like AWS, an internal data center, Ali Cloud, Azure, Google and Open Stack and VMware. So there's a bunch of infrastructures which we essentially abstract away, then project Gardener is really your friend. And we'd see that actually, that uptake in our open source community as well. There are some members in the community who have nothing to do with SAP software which is quite interesting and refreshing to see as well.

Karsten Hohage: OK. And how would they get into all this if they don't primarily have anything to do with SAP?

Vasu Chandrasekhara: Project “Gardener” in the beginning does not have anything particular to do with SAP or SAP business software. It's a Cloud Native Kubernetes Management Tool that can be used in any type of context. So we see an uptake in our community of start ups which essentially have the same challenges around creating homogeneous clusters across different environments to make them secure. And that's essentially what Gardener offers.

Karsten Hohage: OK, that was maybe my bad that I didn't quite understand that at first, but it was maybe also good to point that out again, that it's not at all necessarily specific to SAP what Gardner does. Some of the things you said made me think. Now, Gardner clearly seems to solve a problem that quite a few people, quite a few organizations out there have. We usually do things in a proprietary way, especially if this solves the real existing problems. Why are we doing this open source?

Vasu Chandrasekhara: First of all, I think you can say it's about the recognition of the power of open source. So, yes, the Gardener team, when working with Kubernetes, immediately profited because Kubernetes and all its ecosystem projects and code was created in open source as well. And that facilitated that our project became very fast and the development was quite robust because we were standing on the shoulder of other projects. But as you mentioned, this does not answer why would SAP open source a tool that clearly solves an enterprise requirement? But again, there's a spoiler alert when I, and we, initially argued for creating the project, also as open source. It's undeniable, I guess, that developers, and with them the entire industry, gravitates towards open source. We chose Linux. We chose containers and Kubernetes, exactly because they're open. And we could bet on the network effect, especially around skills. I

mean, Tim mentioned that we built Gardener not only to manage Kubernetes, but we built it using Kubernetes. The way how we decided upon doing project “Gardener” in Kubernetes and so on is currently replicated at thousands of companies and most of them also our customers with the same result because, you know, all companies are currently, digitally transforming.

Vasu Chandrasekhara: They're becoming, to a certain extent, software companies as well. And in many discussions we had, we recognized that they have the same challenges that we are solving with Gardener and this was, in essence, a key eye-opener. And our leap of faith was that essentially, if you develop Gardener in open source, we would be able to have completely different conversations with our customers and, precisely, with anybody interested. We just had to put a strategy around this core belief. And coincidentally, if you look for that, that was essentially also the same timing when the Open Source Program Office was launched as well. Yeah, and last but not least, SAP, and this was an advantage for us, differentiates with business processes and planning software. So that on top of the layer of what Gardener actually does. We made the case to develop Gardener completely as open source, and that not only opened the gates for adoption, but also for co-innovation and feedback loops directly in code.

Karsten Hohage: OK, I want to jump back a little bit to what you said, Vasu. You said the whole world is seeing that the gravitation for developers basically goes towards open source. Now we have Tim here with us as the hands-on guy. Can you confirm that?

Tim Usner: I can definitely confirm that. As a developer, the Kubernetes project shows what was only possible to achieve with Gardener, with Kubernetes being the open source project, right. So I talked a little bit about the Kubeception model and I guess that never had occurred if Kubernetes wouldn't be open source. Because you have that huge community, you have that huge amount of know-how that's available. A lot of people you can talk to that are open towards your technical concerns, technical questions, and also, of course, technical ideas to help you with your implementation and to strive towards a common goal. And I think this is also something that Vasu brought to the table. So that common goal is, of course, a common goal of different organizations, companies, even individuals. And we all work on the very same projects to achieve this

goal. Right. So you have full man power, full knowledge and from different companies, different aspects. And this is only possible in the open source world as I experienced it.

Karsten Hohage: And more specifically, maybe also to you now Tim, from a developer perspective, if you were not seeing the world from within Gardener, but basically as a user of Gardener. Why would you use something like Gardener? Why would you use Gardener instead of something else? And what exactly is a developer's benefit from Gardener?

Tim Usner: So, I mean, Gardener solves a really common challenge, I would say, or a common and tough challenge. So…

Karsten Hohage: Does it relieve you of, like what do you not have to take care of anymore with Gardener around?

Tim Usner: Gardener is not bound to any specific infrastructure provider and all the different infrastructures Gardener covers with its extensions are treated in a really homogeneous way. So let me maybe explain from a technical point of view what you get with Gardener. Simply put, Gardener offers you a vanilla Kubernetes cluster on various infrastructures or hyper scalers and also the Envers covered as well as Open Stack. We have bare metal with metal-stack.io and even further providers like Telecom, OVH, Equinix packets and so on, you name it. And what Gardener does is, it takes care of the day one and day two operations for these clusters. This means that Gardener is not only capable of provisioning or deprovisioning thousands of clusters but also observes and auto-heals your clusters, if necessary. It upgrades components in a rolling fashion, for instance, if there's a new Kubernetes or worker OS version available and scales different aspects of your clusters, for example the control plane but also the worker nodes up and down, depending on the workload you put on that very same cluster. So it gives you all these kind of managed features. And the interesting point here is also from a technical point of view that these features might sound familiar to one that already worked with Kubernetes .

Karsten Hohage: Because you said that Gardner itself is built on Kubernetes, right?

Tim Usner: Absolutely. So features like the rolling updates, self-healing, auto-scaling, these are all features already implemented by Kubernetes. And we thought, why not take advantage of these features and manage the Kubernetes clusters with Kubernetes itself, as I said, also called Kubeception. So let me do an example because this sounds kind of abstract. If you are in a Gardener managed Kubernetes cluster, you never see the master components of this Kubernetes cluster. For example, the cube API server, the cube controller manager and so on. And why is that? Because these components are provided as a service to you, posted inside another Kubernetes cluster which we call, by the way, a seed cluster and actually a seed cluster can host hundreds of other control planes of other clusters. And on top, further assets like the backing machine of your worker node are modeled as a Kubernetes resource in that seed cluster again. So bottom line, we take advantage of all the great features of Kubernetes for managing further Kubernetes clusters. And another important feature of Gardener, from a technical point of view, is that even though Gardener brings these Kubernetes clusters to the various infrastructures, it only uses the very basic resources of these infrastructure providers or infrastructures like the VMs, routers, load balancers etc. Thus, you can really quickly extend the scope of supported providers and the user experience is very homogeneous at the same time, no matter where the cluster runs. On top of that, Gardener also exposes these cluster services that I just talked about via an API in the form of an aggregated API server, as well as a web UI for your fleet management.

Karsten Hohage: OK, I hope there are some developers listening out there because you lost me there in some places but I'm sure a lot of people listening will have caught a lot they can benefit from. My simple understanding is as a developer, I just don't have to worry about my infrastructure provider anymore.

Vasu Chandrasekhara: Maybe I can take a little bit of a different viewpoint, you know. Remember, Kubernetes for whatever workload on top is not the end goal. It must serve a purpose for users or stakeholders. And with Gardener, we ensure that this purpose is mainly highly safeguarded with technical means. Maybe the other way to look at it is, like, imagine your business team. You've created a modern solid cloud native application or service, fully scalable in containers. And now you've got that running and all of a sudden from your business requirements say, oh, you've got it running on AWS, but you also have to get it running on Ali Cloud because of China or an Azure or

whatnot or on premise. And then you, as your development team, did everything to ensure that the workload is containerized and portable as much as possible. But now this challenge to go from one provider to another is actually quite difficult because all of a sudden, you know, you have dependencies in infrastructure, not with respect to the differences of machines and so on but the version of Kubernetes, the roadmap, the deprecation plan and so on. So you have to manage that all of a sudden or otherwise you're completely dependent on the roadmap of someone else who's providing you that. And if you think about it, you had to suddenly qualify your product on different types of managed Kubernetes offerings. We've done this and we've experienced this at SAP already, your total cost of development will explode. And therefore, with Gardener, you get a single pane of glass to roll out homogeneous clusters. You are in control of your version. You can do your conformance testing and you have a single roadmap across all of the supported providers. And last but not least, let me also say, this is not something we are observing but we are also seeing this in the cloud native ecosystem. That people who have serious business workloads running on Kubernetes will say: "Oh, having the managed control plane behind an opaque wall, where you don't have control anymore, is a little bit problematic". So serious workloads need some type of an access, shared access and visibility and with all the tuning options of the control plane to safeguard the service.

Karsten Hohage: So, if I want to create a short answer, like I sometimes do that for myself: From what you were saying, Vasu, that is what first looks like solving developer issues also solves quite large business questions, as in being more flexible with your infrastructure provider or not being as locked in, not being as dependable on versions of whatever, etc.. Let's maybe turn to another perspective here: We've now talked about the use of Gardener on top of Kubernetes. How about contributions? I mean, this sounds like this is not just one service or something. Sounds like, at least halfway, if not very complex project. How difficult is it to participate?

Tim Usner: Gardener is squarely located in a domain where infrastructure, cloud-networking and Linux knowledge, and probably more, meet. So that's exactly the domain knowledge of a common Kubernetes admin, I would say. A DevOp and/or somebody working in an SRE role. And here is the funny thing: As you remember, I said, we lifted our nifty secret that Gardener was built on Kubernetes. And, well, that equation even continues because the Gardener architecture matches Kubernetes in

many areas. For example, Kubernetes has a cube API server, Gardener has a Gardener API server. Same for the scheduler, the controller manager and Kubelet which we call Gardenlet. In essence, if you understand the Kubernetes architecture and the Kubernetes way of solving things, you immediately understand Gardener. So that, for example, means if you have the skills in Kubernetes and know the concepts of controllers, CRDs, watches, finalizers, deployments, all the different Kubernetes native objects and so on you can easily transfer to the Gardener world and actually start contributing to Gardener. And so this is not only theoretically spoken but this also is exactly the feedback that we got from many community members who already contributed to Gardener.

Karsten Hohage: OK, I get it. That sounds like an approach if you've understood Kubernetes, go start work with Gardener, right? So what's currently the most active subject in the community around Gardener or where do you see the most active contributions?

Vasu Chandrasekhara: Maybe I can take that, Tim, even though you're on the day-to-day basis working with them. I think, you know, in the beginning when we started the community, we really had experts in the field joining our community. They didn't need much documentation and so they could read the source code. But what we're seeing now and what is most active in the sense is that, you know, new people are coming in. And the first question they have is: ‘How do I set this up?’. It's about the installer and about documentation. Even though these beginners are maybe into Kubernetes and then can easily transfer the skills into Gardener, and if that happens, all complexity falls into place and you can kind of make the connections yourself. But beginners that now are joining the community without much prior knowledge, they have quite a high ramp up of knowledge I would say. And that's a pain point. Our experts directly ask questions about documentation not being up-to-date and so on. That’s, I would say, where we are seeing active community discussions and so on. If it were a start up, I guess we would have prioritized, you know, the full-fledged installation experience with the UI, with documentation at first. But we kind of prioritize, as a good old German company, on the functionality that you get with Gardener and now we need to catch up. That is one part. And the thing is that we are reinventing the way how we are setting things up because within SAP we do development, we have a pipeline and we transport all the deltas and changes from development to quality to production. And that is currently being

reinvented. In the last community call, sometime in the summer, we gave an overview of what we are building there. It's called the landscaper. And with this tool, we want to make that available to our community. We'll not only be able to install a full Gardener landscape but we will be able to streamline patches, updates and upgrades with the landscaper into foreign environments, you can basically attach yourself to a release train. And that's, I guess, currently the most active subject. There are many other aspects like spot instances and ARM support and so on at the fringes. But I would say that is the most active thing, which also is connected to our adoption question.

Karsten Hohage: Speaking of participating, where do people go as a first stop that are interested in Gardener?

Tim Usner: Very good question. So a good place to start is our website, of course, gardener.cloud, where you can find further documentation, articles, blog posts, links etc. So this is the single point of contact, I would say. Then, of course, we have the Gardener organization on GitHub. It's GitHub.com/gardener and there you can find various sub-projects for the whole Gardener project which is especially interesting for people who want to try out Gardener. There's a project called Gardener setup and this is a simple set up routine which helps you set up your Gardener environment in your very own infrastructure or with your very own infrastructure accounts. And last but not least, we have different communication channels. The links are also on the web page but I especially want to mention that we are on Slack in the public Kubernetes workspace. The channel is called #gardener. So a lot of our peers are there, available as well and happy to answer your questions and collaborate with you. We have a bi-weekly community meeting for which we are always happy to see new attendees. So eventually looking forward to meeting you there.

Vasu Chandrasekhara: Maybe to also mention that in the GitHub organization of Gardener, there are a couple of projects, sub-projects which are used by community members without Gardener, like the DNS manager or the backup component for etcd. There are community members who are using those components without using the full-fledged Gardener.

Karsten Hohage: OK, great, then let's leave it at that, because, as I said, we try to stay in a little bit of a predictable timeframe here with the podcast. So what is usually my last

question is, if you could wish for three things for people to take away from this podcast, the three key takeaways, which three would it be? Maybe you want to take turns. I don't know how you want to do this, Tim, Vasu.

Vasu Chandrasekhara: I guess, the takeaway one, I would say it's about open source having reached the boardroom. I mean, it becomes very clear that open source, even if it's not anymore, or was never, just about freely using available libraries, components or tools to optimize your own software production, open source is strategic. It has also reached our boardroom. It's about getting into a new kind of conversation with your customers. And maybe I'll do number two as well. In my opinion, you know, Gardener and what we're doing currently with open source is about solving concrete challenges by co-innovation because open source, in my opinion, democratizes the innovation process. Because now all of a sudden, users of a particular product or service, they vote and decide increasingly for open source variants, such as with project “Gardener”, because that allows them to freely innovate and solve their concrete challenges by developing exactly what they require because the code is now open. It's all there. And this is all of a sudden a user-centric process that has tremendous advantages. And it clears out the middleman. Lastly, if others start using and contributing to your innovation that you put out there, it allows you to secure your investments for the long term.

Tim Usner: So point three is left, and I'm happy to take this point: Of course, addressing especially the developers, operators, but also end users of Gardener. So I clearly want to point out that Gardener solves problems by applying Kubernetes and Kubernetes principles itself and especially developers and operators who are familiar with Kubernetes, they will notice this concept and they can contribute, by the way. They will understand Gardener, they don't have to learn any special knowledge, anything special we brought up. It's all about Kubernetes eventually. And that's really the key takeaway if you use Gardener and develop for it.

Karsten Hohage: Nice final words there. So at this point, I thank you again, Vasu. I thank you again, Tim. It was great to have you here.

Tim Usner: Thank you for having us.

Karsten Hohage: All right. And thank you all out there for listening to „The Open Source Way“. If you enjoyed this episode, please share it. Don't miss the next one in around two weeks and subscribe to us on Spotify, Apple podcast, Google. We're not only on the openSAP site but on all your regular podcast channels and you will hear from us again.