Rich Savage: Thank you for joining us today. I'm Rich Savage with the Red Hat Team at Carahsoft, and we're sitting down today with Chuck Svoboda, the Lead Transformational Specialist with Red Hat to talk about the role containers play in IT modernization.

Rich Savage: Chuck, to get us started, give us your definition of containers.

Chuck Svoboda: Containers are a standardized packaging of apps, databases, whatever you could think of, that are portable, immutable, and provide isolation from each other. In a lot of ways I like to think of containers as kind of like the new-school RPM for the cloud that can really run anywhere and provides homogenous operational experience, at least on the outside.

Chuck Svoboda: One thing that's important to understand, because I think there's a lot of confusion on it, containers are not an evolution of or a replacement for VMs in a lot of cases, because the UX is so different between a VM and a container, you really need to treat them a bit differently. That's actually where I think a lot of folks get themselves into trouble, which is why I talk a lot about that. I want to make sure that we bring it up directly, but Red Hat can help you with that.

Rich Savage: In your view, what is the difference between the virtual machine experience and the container experience?

Chuck Svoboda: Well, if you've heard of the comparison of VMs to containers of pets versus cattle, is that a VM is traditionally very stateful, very persistent. Operators brag about how long their VMs stand up, whereas containers are supposed to be very ephemeral. They can spin up and spin down whenever they want with no interruption to continuity, with no interruption to service or anything like that. Let's be honest, it's a gross oversimplification but that's a big piece of it, and then so when you're building applications or services to go in a container, versus in a VM, there's a lot of operational and development concerns taken into account.

Chuck Svoboda: One great one that I always bring up is configuration specifically of end points and things like that, like hard-coated IP addresses, which is bad practice in general, but those absolutely do not fly in a containerized world. Again, there's many more that Red Hat can help you work through, but that's a huge difference, one that comes top of mind every time as I explain to a customer.

Rich Savage: What are some new and interesting facts around containers?

Chuck Svoboda: Containers have traditionally been thought of... useful for, rather, cloud-native web applications. Think of like Node.js apps or something like that, kind of greenfield development. Where we're kind of going into now is as I really kind of think about Gartner Hype Cycle, we're kind of coming out of the trough of disillusionment. We're now applying because those cloud-native apps are so few in number compared to the larger swaths of use cases such as high-performance computing applications. Applications that need to use GPU. Stateful applications and services like databases and message keys and things like that.

Chuck Svoboda: Edge computing, which is a really interesting use case for containerization, and then finally, we do actually understand that VMs will have a place in infrastructure for quite some time longer. While, containers are actually a great way to run traditional VMs. You get the management aspect or the benefits of the management aspect around single-pane glass or unified control plane for all your containers and your VMs, so just run them all the same, basically.

Rich Savage: From your experience, what is the current state of public sector IT modernization?

Chuck Svoboda: Sure. I think the simplest way I can say that is we're just scratching the surface. A lot of the conversations I'm in start with, "Hey, we want to adopt containers", which I'll be the first to tell you... I'll probably say it part of this podcast a few times that containers by themselves really don't get you anywhere unless you modernize your methods and practices around the same. I think a good swath of public sector is still just focusing on, "Hey, let's put something in a container and call it 'modernized'". There are quite a few agencies out there, particularly in DOD and some of the civilian like USCIS that are much further and understand that true modernization, not only building new applications better but modernizing existing legacy through the use of containers as a tool, is the right way to go about it.

Chuck Svoboda: I think we have not fully rounded the corner, though. There's still so much more to do and we're still a long ways about it, but some very exciting journeys. There's a lot of opportunity for agencies out there to do what they need to do to truly modernize and do things the right or new way.

Rich Savage: How can containers help with IT modernization?

Chuck Svoboda: As I said, containerization by themselves doesn't get you much without adopting new methods and practices, specifically think about agile methods and practices to modern application delivery. Just going out and buying a bunch of containers and throwing them at that, including your culture out of these organizations doesn't get you very far. However, from a technical aspect, the things like the isolation, the immutability, the homogenous development and operations experience really enables us to... or actually lowers the bar to adopting some of these methods and practices.

Chuck Svoboda: Think about like Red Hat OpenShift as a container-based application platform with all of our development and operations and opinions on top of it to move fast safely, securely in its scale. Containers are definitely an enabling implementation detail of that, and so I think at the end of the day, yes, containers are very helpful. They allow all of this to happen, but really, a developer should never see a container in a modernization experience. They should still just able to check in their code and let the platform handle the building of it, the securing of it, the testing of it, everything else that needs to be done in it in a modern CSE pipeline, and then just so happens that that all happens because of containers.

Rich Savage: Do agencies understand the distinction between modernization and containers?

Chuck Svoboda: Yeah. I think actually it's funny. As I said earlier, containers don't really matter at the end of the day. They're an implementation detail. They're a really important one, so you need to do them, but funny enough, I get brought into modernization discussions simply because they want to talk about containers, and then that opens the door for me and my friends in the industry to kind of get in there and have a real discussion about what actual modernization looks like.

Chuck Svoboda: A couple of examples, though, for instance are it was announced at Red Hat Summit a few weeks ago as Lockheed Martin is modernizing the F-22 Raptor, basing facing speed delivery to get functionality, features, capability to the actual jet, the actual weapons system from four years down to one. We're doing that with Red Hat OpenShift as an application platform, a development platform, using agile methods and practices derived from Red Hat Open Innovation Labs as well as our traditional consulting program at scale. No, we're not putting containers or even OpenShift for that matter, on the airplane yet because we're just not there yet. We're not ready to do that yet, but containers are a great way to accelerate the development and provide development guardrails for the development teams to move fast and rapidly iterate.

Chuck Svoboda: Another great example is USCIS with basically their eDocs line of business where we're actually completely modernizing and revolutionizing the way that all the documentation around immigration is being used, which is a very sticky subject right now in the America we live in. It's really cool to be able to be a part of... again, this was announced at Red Hat Summit a few weeks ago. A really cool part of providing a better way of life and a way of doing things on a very stressful situation for folks who are trying to get into our country and actually act so that's pretty neat.

Chuck Svoboda: Again, containers aren't just an implementation detail. They are kind of I think of a fuel as part of a modern application delivery lifecycle or pipeline, but they're not the reason, they're not the catalyst for it all happening. The last one I'll say example-wise is a radar system that detects ICBMs. That's a really cool use case that is really important to national security that we're leveraging Red Hat OpenShift and our consulting organization to again leverage the platform and everything to move fast, to do rapid iterations... to fully modernize and get this agency off of old, unsupportable mainframes that are just... really have nowhere to go anymore.

Rich Savage: To sum up for our listeners, why Red Hat OpenShift for public sector?

Chuck Svoboda: Given that this is public sector and there is all kinds of security and controls and accreditations and things like that we have to worry about because in public sector lives literally depend on this stuff, I like to start with security. Red Hat OpenShift runs on a battle-hardened operating system, Red Hat Enterprise Linux, and if you think about containers as a whole, containers really aren't that special. They're basically Linux kernel primitives. I've been saying it. Again, they don't really matter, but they do in terms of they need to be done right. Everything or the way that you run containers through Red Hat OpenShift inherits a lot of the security controls that are baked into Red Hat Enterprise Linux.

Chuck Svoboda: Anyone who's been looking at modernization through containers, really you can't throw a stone without hearing about a CDE or something that was exposed to some sort of exploit and there's several this year that actually came out in the past year that Red Hat OpenShift just wasn't affected because the other layers of security that we bake into RHEL that, again, OpenShift inherits, it's just not affected us. It's a basic completely moot point. Basically the exploit that happened inside of a container just couldn't break out because of the awesomeness of the defense in-depth it provided through RHEL that again OpenShift inherits.

Chuck Svoboda: Another thing to remember, too, is OpenShift is Enterprise-grade Kubernetes. Again, if you're doing containers you're probably going to do Kubernetes. The other container management platforms that have been out for the past few years just kind of don't really matter anymore. Everyone has kind of gotten bored with this now with Kubernetes, and so understanding that Kubernetes isn't an app platform by itself, it's basically incomplete. You need to do a lot of other things to make it functional for production. Things like telemetry, logging, metrics, monitoring, all of the CMCS, things like that. An even easier or a better way to do management through like a console both operationals and dev and provide an easier development operation experience.

Chuck Svoboda: Well, OpenShift comes with all of that baked in, so you're not hand rolling it, DIY-ing it, having to support it yourself on the Kubernetes platform. A long time ago, I remember... I think this was maybe not a long time ago, in IT years I guess it is, but 2016 I remember a conference it was like... I remember someone said, “If you were to roll your own platform with Kubernetes, hopefully you'd come out with something that looks OpenShift", and I still believe that's true to this day.

Chuck Svoboda: Another thing to think about, too, is that what Kubernetes is and what OpenShift is, is a gluing together of a lot of really important open source projects. Each one of those open source projects has its own lifecycle, has its own set of updates as they come out, and the management of that is really, really hard. Well, with Red Hat OpenShift and part of our trusted software supply chain that provide from an open source perspective and curation is that we take everything in the upstream, compile and package it in such a way that makes it really easy to install and also maintain both day one and day two operations so you don't have to worry about all that kind of the actuating that has to happen with running a Kubernetes-based platform.

Chuck Svoboda: Lastly, regarding kind of the operations and support is that there's someone to call if you need help. OpenShift and any other Kubernetes platform is literally like installing the cloud. Its compute, its storage, its networking, and a whole bunch of other stuff that is frigging hard and you don't want to do that on your own. No matter what anybody tells you out there, unless you're running a managed service, your data center does not look like the reference architectures for any of the upstream open source projects. Calling Red Hat, we've seen it basically all. We can absolutely get it integrated for you and help you support it.

Chuck Svoboda: Take it up a level is that by leveraging Red Hat OpenShift, you get access to our consulting organization that has decades worth of agile software development, skills built in from building software. Some of the world's trusted software for Red Hat for our customers, whomever, and bringing agile methods and practices such as Lean, XP, Scrum, Kanban, et cetera, domain-driven design, test-driven development, pairing, et cetera, and that's how you get access to it so that you can take those prescriptive CI/CD pipelines that we ship with integrated with all of the other trusted partners and IOC community to deliver applications reliably, repeatedly, rapidly, and with a good ROI.