The i4 Group DevSecOps in Cloud Native Product Development

Tue, Oct 11, 2022 9:45PM • 59:39

SUMMARY KEYWORDS
security, application, kubernetes, secured, architecture, ensure, people, industry, kubernetes cluster, micro services, devops, code, environment, running, organization, deploy, demian, cluster, security policies, cloud

00:00
All right. Thank you very much. So good afternoon, you all, we're kicking off our webinar for today.

00:10
Today, your hosts are myself, Charles Maddox and Dr. Demian Higby. And we're hoping to give you just an overview and a taste of DEV SEC ops in cognitive product development environment. So we have a small group today, it's our first webinar series, but we're hoping this is going to be we're going to have a series of these over the next several months, which, you know, hopefully can meet some, you know, very interested in topics that that you find valuable to take back to your, either your academic or your work environment. And, yeah, we can definitely like to hear your feedback too, at the end of this, just to kind of give you an idea of what our series of webinars is all about. We're hoping to bring some scholarly and, you know, research information on these topics, as well as what's being applied in industry, in practice. So, you know, combining that that nice mix of what's kind of on the cusp of the research topics, and what some of the research projects have actually shown to produce, and actual industry practices that we've seen in industry. So to go into who we are our house. So I'm Charles Maddox, I'm founder and principal of the eye for group and we are a consulting and training company.

01:34
Myself, I've been in the software development industry for the last 25 years. So I've seen a lot in some of the transitions, moving from more traditional ways of working into agile and DevOps and now dev SEC ops, and I'm actually a research student as well. University of North Texas,

01:53
actually studying DedSec ops, and I even called Dr. Demian, my mentor, he's the front runner in the Dallas Fort Worth industry. We're both here, local, Dallas, Fort Worth

02:06
residents. And yeah, I'd like to turn to Dr. Damon yet to give a quick intro to yourself.

02:14

Sure. Thanks, Charles. Okay, so hello, everyone. My name is Dr. Demian, Ebay. I have a PhD in computer science now for about 10 years. And before then I've always been in it. So I tell people is the only field I've ever worked in. So it's kind of like my life experience that I bring on board whenever I'm in a panel or any kind of discussions like this. So hopefully today, we have some value to give to you guys for attending. Thank you.

02:48
Thanks, Dr. Jamia.

02:51
So yeah, as I mentioned to you know, both Dr. Demian, and I were both industry practitioners, and scholars and students, I don't claim to be a true scholar like Dr. Demian, quite yet. I'm still working on it, but doing my research projects in this area, and learning a lot. And actually, you know, very fortunate to apply some of the things that I've been working on the last 25 years and actually research these things in depth. So, overview of what we're going to cover today is I'm gonna talk first about what is industry 4.0. And industry 4.0. It's kind of a, you know, it's kind of the tip of the iceberg on why dev SEC Ops is getting so popular these days. Then cover dev SEC ops at a high level, you know, why dev SEC ops? What are some of the problems and in the industry that are driving the need for dev SEC ops, and they cover some of the processes and practices that we've seen,

03:44
you know, a lot of companies trying to adopt, and what the research is suggesting are some of the common approaches and common pitfalls and things that everybody's trying to adopt. And so kind of looking at the best case scenario, what are these core practices that principles of operate operating a good dev SEC ops environment? What does that look like in laying that foundation out? Then Dr. Damien's gonna take us through a use case scenario where actually using a Kubernetes cluster and actually show some of the various levels of security and how they're implemented in in actual code, and we'll do a simulation on that. So you can actually get a taste of what that would look like building a cloud application. And then last but not least, we're going to talk about you know, if you're interested in learning more about dev SEC ops and kind of what you know, what might be a learning path or an approach to kind of get involved in this industry. We're going to give you some information on that as well. All right, since we got a small audience,

04:42
I would say let's at this point until it till it's actually necessary. You guys could raise your hand use the reactions button and the in your your your panel, your zoom flooding panel. You can use the reactions, your reactions button to

05:00
Raise your hand and you can ask questions real time since we have such a small audience, we can make a collaborative that way.

05:07

If need be, if you think it's not, you don't want to do that we are monitoring the chat as well. So type your, your question in the chat, and we can answer that question

05:16
when the webinar is over towards the end of the session here, okay. All right. With that said, we'll move on.

05:25
So, industry 4.0. So let's talk about a little bit about tech. So the term industry 4.0 has become popular

05:35
in such that we have a lot of technologies now that are being created in the cloud and being driven from the cloud architectures in our, in our current workspaces, right. So just think about, you know, cell phones controlling, you know, start your car, think about, you know, thermostats, refrigerators, connected to your internet, I mean, just the, I put a

06:04
pure water filter on one of my faucets. And I can, I can monitor that through the, through the internet on how many gallons I'm using. So more and more. So these days, we have technologies, communicating through cloud infrastructures, that are really driving the new economy, there's so many opportunities in this new economy. For that, I mean, contrast that to, you know, looking at the previous industry, you know, going way back to the industrial age, mass production age. And then more recently, you know, you got some, you know, your automation and computer age, that that predecessor, predecessor to our current information age, or you call it industry 4.0, there's very distinct ways of working as well, in those previous industries, these previous revolutions, we'll call them. And to that point, though, you know, if you think about it to the management structures, that were in place where these different industries and different revolutions were aligned a certain way for those industries, for those in those those revolutions, those types of industries that were in previously. So what we're facing right now, in today's space, is, is not only just a blend of new technologies, but a new way of applying management structures and constructs around this, these new technologies that we have very fast paced, you know, a lot of feedback, very cross functional, a lot of new, a lot of, you know, the Agile revolution, the DevOps, you know, the DevOps ways of working, these are the things that are, are, are necessary to kind of keep pace with these technologies and to pay for these technologies, how fast are moving, and then you take security, and you align that with the need for security changing at such a fast pace, along with these new technologies. Now we have the perfect blend on how do we get, you know, the management constructs aligned because people don't not want to have their their businesses at risk, you know, for breaches, vulnerabilities, etc. So it's a serious thing. And so that's why DevStack Ops is really coming into play pretty strong as how do you change the culture that the management constructs around this new way of working to make sure we're keeping pace with keeping our organizations and our applications and our product secure, and still able to accelerate at the, at the pace that the market wants us to accelerate at? Okay. So you know, as mission. And just as I just mentioned, you know, the IoT space is pretty significant. When you look at how the different interconnected applications and you know, endpoint technologies especially, may be opportunities for breaches or vulnerabilities. They say 80 to 90%, I think I think it's 90, don't quote, some various studies, but mostly 90 proximately 90% of

data breaches and security issues, reside at the endpoint, reside at the user on how users are using various technologies.

09:22
Most most of the time the

09:25
we're doing as a development organization is our security policies. They're in place, but they're not being used properly. They're not being used properly. So a lot of a lot of these security

09:38
aspects come down to how the end user is actually using some of these technologies and devices. Okay.

09:48
So, let's talk about when it comes down to when we're trying to put in these policies and procedures and new ways of working. Let's talk about what it what it looks like actually on a development team and

10:00
worry, or an operations team trying to build, enhance or add new services or policies

10:07
into a new product, and how that leaves insecurity, maybe

10:14
on the outside looking in, and DevOps, they call this, the, the walls have confusion. And they even have another group here, that's not represented, but they call it the business, you know, the business intent, the business wants something new, put into production, some new feature or enhancement, that needs to be given over to the development team, you know, there's a wall there, you know, this is different, this different swim lane, you call it a different silo part of the organization. And so the development team is puts that that feature or new enhancement on into production, and now it's the operations team to build any issues or any feedback that they may get from the field for monitoring and, and any type of system that or usage that might be going on in the field. But again, different part of the organization, it's kind of a wall in a silo, potentially, not ideally, obviously, but I'm not, I'm not promoting that way of working. But that's what might be going on in our organization is different silos, trying to communicate to one another. And then you have this other group, you know, so now now we're talking about web based technologies, we're talking about IoT, we're talking about a lot of, you know, remote users in our system. And we're building new products like crazy, right, in that space, and security becoming more and more of an issue. But then, as the picture shows, here, they're on the outside looking at another another wall of confusion, how does that security group get its policies and controls and, and any type of compliance and things to make sure that we're actually building products that are secure, so that when users are using these devices and applications on various devices, within our organization, we do have a good understanding of how things should be handled. And so they're a part of the conversation right from the beginning. So in a perfect Dev, DedSec, ops world, all of these groups are kind of working

together and EC pi seen this diagram before this little infinity symbol representing DevOps. But now dev sec, sec security across all of these aspects of your traditional software development lifecycle.

12:36
And so, you know, that's that's the, that's the goal is to try to get this level of cross functionality, or at least knowledge share, and understanding shared responsibility happening across the entire group of those trying to produce this new new value going out into the into the market. So dev SEC ops, and the principles that apply to it.

13:05
More so follow the line of principles around DevOps. And you can see the core DevOps principles along the left side there of culture, automation, lean, measure and sharing. But it takes on a different twist, each one of these, each one of these principles, as a slight twist to it to make sure that security is being considered. So let's take number one culture, a security minded culture, that we understand that breaches can happen, we understand that our products are susceptible to you know, and are vulnerable to attacks. And we were taking measures to to counteract that. So it's threatened mind center on all of our developers and our operations, folks that are those doing the work and on a daily daily grind, that they understand these these concepts, automation, that our automation tools, can take the initiative to actually automate some of this security, testing, security scanning, that we're making sure that wherever possible, just like in DevOps, to kind of create that that flow of value is going to optimize in the most optimized way that we're considering how to automate wherever possible. And then, from a lean perspective, one of the things right off the bat in any DevOps implementation, just like dev SEC Ops is need to understand our value stream, how does value get out the door from you know, from a business, ask all the way to production and understand those steps in the value stream and understand where are the points in which you need to be applying security and making sure security is being considered in these various aspects of the value stream. So very, it's a tool, actually a Value Stream Map, to look at how we're actually mapping our security policies and our security approach to the

15:00
The organization so very good tool that we can actually take from lean, and then measure as any any DevOps implementation, being able to measure the from development all the way through to production and monitoring, we need to make data driven decisions. And data should be driving how we actually react to security as well, giving us the opportunity to proactively make security

15:26
changes or enhancements or putting security policies in place, by the simple fact that we have a data driven approach to our dev SEC ops pipeline, and then sharing the sharing of information that you know not, that's one of the actual challenges I'm going to share with you share with you in a few slides is that typically, there's a very small fraction of individuals responsible for security in a typical organization versus those building and enhancing and supporting new products. It's up to that small group of you know, security, focused individuals and organizations to share all this information about what how do we need to secure our products? How do we need to add security policies into our, our development pipeline, and it's a big task, you know, this, there's a small percentage of people that have

this information about the security concerns and what's needed. And there's a lot of people that need to use and apply these methods. And so being able to make this information shared broadly and widely and that and internalize into our processes is a big challenge. So we need to be able to share and see that shared responsibility as everyone's responsibility, which is security. So

16:42
this is right out of a research study, and I'll provide the when we provide the

16:51
notes to the study will provide a link to the actual study in the reference. But it's a very good, very big picture, I thought that shows how the different security controls may be applied to the different workflow stages within your development pipeline, a lot of these different security. So let's take the very first segment, on the far left, in your continuous planning and development cycle for the practices, their continuous planning, continuous exploration, and getting new, new backlog items into your into your team's backlogs. And how do you plan and deliver on so many things, so things that we could actually be applying from a security controls perspective, like, you know, Threat Modeling practices, you know, architectural designs that account for security, certain guidelines and controls, to make sure that we are putting into our IDE from a developer standpoint, going on into the build the build process, we're actually putting in, you know, applying development practices that align with our security policies that are needed, you know, you know, unit testing some of the security practices around that, you know, infrastructure, code analysis, static code analysis, security scans. And so all the way through the entire continuous delivery pipeline is some, you know, there's some controls and some practices in guidelines, we might think that might apply for the products that we're working on. So very good, a very good model here to kind of analyze and look at how this might be applied in an actual environment. And Dr. Demian, is going to give a simulation where we're gonna touch on some of these areas, you actually can see some of these in play and how they may work in an actual cloud development scenario.

18:43
So,

18:45
as mentioned, you know, one of my research topics, I'm actually in my PhD studies in UNC, we're doing a lot of the cultural dynamics, focused studies on why is it hard for companies to, you know, to embrace their setups? Like, what are some of the key barriers, and some of the cultural areas that stick out?

19:06
Typically, are the collaboration, you know, I'm just this list here is wrong, the cultural areas, collaboration groups just don't collaborate well together. You know, again, security is typically a group, like the outside looking in, or they are the cops of the organization. They don't look like they are truly a part of, you know, value delivery, as a developer would be, but they're the guys that will get you in trouble. All right, so they don't you know, that we don't want to collaborate with them as we will collaborate with our colleagues on the development team. Again, knowledge sharing, as I mentioned, that there's a small fraction, typically of security professionals in the organization compared to the masses of developers and testers that need to get this knowledge they need to embrace it and have

some understanding about how to have that security mindset within the organization so that we can you know, react to security breaches, and

20:00
proactively put in controls to, to, to address these things before they happen. But that requires the knowledge to do so. And in some of the the the opportunities, there is a challenges and opportunities, training programs, how do we have an ongoing communication

20:16
messaging campaign within organizations that security is primary, that it's just a part of the way we do business, it's, it's our everyday look at things, because we're in this type of environment that, you know, it's very risky, or, you know, people are always looking to attack feedback from the field. And this, actually the challenge in non security, or security focused organizations, but just having that constant loop of feedback from end users. And that goes back into the delivery pipeline. And so a lot of this can actually be acquired through automation tools, and continuous monitoring of the production environment. But we also see want to hear from the end users themselves that actually have the security challenges, or they're, you know, there might be some, you know, users in the field that are actually, you know, trying to, you know, hack out our system and make sure that we're keeping it robust, that continuous improvement mindset, that that's something to that that's kind of hard to drive into the culture of an organization that's just used to delivery, delivery delivery, you know, how do we actually slow down a little bit so we can speed up and build higher quality product and really, really look in retrospect on the work that's being done on how we can actually create a more robust security environment, shared responsibility, a little bit, just like I mentioned before, it's everyone's responsibility for for security. And then trust, trust is something that is big in an organization is typically driven from leadership. But trust in that the developers understand and the testers we are giving them the tools and the resources, and the knowledge and the know how, and we're going to trust that they do the right thing. You don't have to have the security cops come in and, and analyze and run audit in your organization.

22:08
Just to keep things safe, that we trust that our developers and testers are keeping the environment safe.

22:15
Experimentation is an area that, you know, we're a little bit lacking in it's how do we continually innovate in our space and still be secured. So this is a challenge that, you know, still being worked on, that gives us kind of a piggyback on trust, that we actually trust our, our technical folks that they can experiment and still be safe, you know, using some of the guardrails and that are that are in place, and we trust them to abide by the guardrails. And we should still be able to innovate in a security environment with that trust in place. And the last but not least, is leadership. Leadership is really driving this, you know, all of these aspects here, because a big part of adopting a def SEC ops organization is the cultural dimension. So that DevOps principle number one, you change the culture of the organization, actually eat you, you actually, you know, culture, you know, culture comes last, right. So that's something that all organizations are driving towards trying to improve their culture, and having a culture that's ready to

adopt these practices and experiment, innovate, and drive continuous improvement, you'll, you'll find a lot of success there. So with that said, I'm gonna turn it over to Dr. Demian. And you can give us some actual examples of what you've seen.

23:37
Thank you.

23:40
Sure. Thanks, Charles, for that great overview of, you know, dev cycle culture. Alright, so, in this presentation, I'm just going to talk about a sample microservice, and highlight some of the challenges you may face once you start to migrate to a microservice based architecture. For those of you who may not be that conversant with micro services, this is the new application architecture that people deploy when they migrate the application to the Cloud.

24:13
You know, now the cloud is the new normal. But, you know, we'll be using on prem environment for deploying our applications. So the architecture of applications in the on prem, we call it the monolith, the monolith architecture. But as we move to the cloud, we have to redesign or refactor these applications to ensure that they are able to leverage the features of the cloud, for example, scalability, we have to ensure that our applications, well scalable in the cloud, also, you know, in the cloud, we have to do pay us to go so one atop our application to be able to minimize our costs so that we don't have to pay a lot of money. So applications have to be redesigned to

25:00
Follow this micro services based architecture. If you have watched if you have used Netflix, you know to watch movies, which I believe most of us here have, then you are looking at a typical micro service based architecture, this

25:16
distributed applications, okay. So basically what happens is that in a monolith, you have to take the code, which was developed and compiled into a single binary code, you have to split the code into different components, different modules, so to speak. And then you are going to develop each one of these components as a single autonomous application. So, once you do that, you have to find an environment to deploy your application. And the most common architecture you're going to find in that cloud environment is going to consist of Docker Kubernetes environments. Okay. So, basically, what I have here on the screen is demonstrating a micro service a very tiny micro service. So here we have a voting application. So let's assume that we are going to deploy to maybe to select our mayor or even the precedent. So we need to use an application for vote for voting. And here we have the voting app, which is going to be developed as a single autonomous micro service based application. And then we have the result, because at the end of the day, we have to collect the result, to see who has won the election. So this is a very simple micro service just consisting of two micro services, just to give you an idea of what a micro service application looks like. But like I just mentioned, in the case of Netflix, you have

26:47
hundreds of micro services, okay, that actually working together to make your movies accessible to you anytime you want to watch the movies. And it's not just Netflix, you know, the big corporations like Google, even Uber, if you have ever taken the share, ride Uber, these are all micro services based architecture. So for sure, this is the architecture of the future, because the cloud is now the new normal. At the same time, even though these applications are good. They're like I say they are beautiful, the cloud, they have a lot of challenges. And most of this challenges we're going to talk about today is going to be mostly from security perspective. So I'm going to leave other aspects of the complexities for you know, some other time, since we're here just to talk about security posture. All right, so I'm going to show you the architecture of how deploying this kind of simple micro service in the cloud, for the security posture is going to look like and, you know, Charles and I, we thought these areas, we should continue to inform people, because security has to be thought of as a as a as a mindset, everyone in it. Now we need to talk about security as a mindset. Because security hacks are real. And I say this, because before I started delving into security, I was in taking security seriously. But once you start seeing what is happening, you have to take security very, very seriously. And in fact, you know, we've been hearing about all these issues about election, you know, my intervention and the rest of them. These are real. Okay. Just last week, I got a letter from my bank, it's quite a Flagstar. You know, it was actually more for my mortgage. And they sent me a letter. Normally, I wouldn't read most of these kinds of letters, but I happen to open this one. And I was glad I did, because they were reporting that their infrastructure was hacked. And a lot of people's accounts have been affected. So basically, they were just informing the customers and talking about things that we may need to do to ensure that maybe if, you know, credit card details and stuff like that have been stolen, that are leased, and the government is aware that, you know, we were part of this hacking and stuff like that. So yes, security is hacks are real. And as we continue to work on these kind of new architectures, we have to be aware, because being being informed is pretty much the first thing that we need to do.

29:28
All right, next slide, please.

29:34
Okay, so let's assume that you have a microservice. Just like the simple one I showed you. You want to deploy this application onto a in a cloud environment. Like I mentioned earlier, the most common architecture is to use Docker and Kubernetes. Now again, I don't know how many of you are familiar with these technologies, but Docker is basically the environment where you're going to

30:00
run your code is a container. Okay, so container is the unit of deployment, you could deploy your code to man back in the day as a physical server, or you can deploy to a virtual machine. But these days, most mostly people are deploying their applications to a containerized environment. And in the micro services based architecture, containers are predominant. This is what almost everyone wants to use. Okay, so now that we have this diagram here is like an O'Neill. Okay, even though it doesn't look like an onion, but it has different layers, different shells, starting from the inner most core, we have the code, the code that the developer is writing to achieve the purpose of what they want to achieve. Maybe that is your Netflix, maybe that is your Uber application. So that is the code, we're going to start from there.

How do you ensure that this code is secured, then you move from there to the next layer, and the next layer is your container that is your Docker environment where you're going to run your code, like I just mentioned, then we have Kubernetes Kubernetes, is going to help us orchestrate our containerized environment. What we mean by orchestrator here is, does take it to be like an orchestra when in a music, musical orchestra, you need to have the person who is coordinating the rest of the musicians or in the shape in the shape, you need to have also the

31:31
the captain of the ship, the captain of the ship is going to ensure that things run smoothly. So actually, the word Kubernetes is from the Greek word for Captain so. So the container, the Kubernetes environment is going to be the one that would coordinate all the different containers that you have in your environment. And that is mostly from the perspective that the containerized environment is complex, like I mentioned earlier, so you need something like Kubernetes to help us orchestrate, coordinate, organize to ensure that our application run as smoothly as it should run. So we're going to look into that. And then the final layer we're going to look at here is going to be the cloud environment. Like I say the cloud is the new normal, and most mostly people are running all these micro services based architectures in the cloud. So all these four layers, we have to ensure that will take care of the security at each one of these layers. And I'm not going to go too deep, too deep into any of them, it just give you an overview of how this kind of an environment looks like. And also, if you look carefully, each one of these starts with C. So you can easily say that this is the four C's of a containerized environment for seats of security of your containerized environment. Alright, so with that overview, we can start with the code. When you write your code, what are the things that you have to make sure that you take care of

32:57
number one is the library that you're using, you know, these days, we work a lot with a lot of open source libraries. And in a typical environment in the typical code, you could be connecting to a lot of other libraries that will help you with one thing or the other, as you run your micro service. And most of these applications, most of these libraries that you're going to be using, they actually well documented by cn, CF, CN CF is Cloud Native Computing Foundation. If you go to their website, CN cf.io, they have something called the landscape, the the CN CF landscape, and they're not going to be able to show you just because I'm not sure so much share my screen. But you can always go and visit it, you see that there are so many applications, I will say more than more than 100 For sure. The libraries that are available for you in the open source community that you can use to build your application. But how are you sure that those applications are actually properly secured, high assured that all the security vulnerabilities in those applications have been taken care of. So as a developer, they can just take those applications and assume that the end the developers have done a good job? No, you have to also do you do your own due diligence in making sure that any security vulnerabilities in those external libraries have been properly taken care of. And if you're not aware, you will not be able to do that. So that is one thing with the code. But also, we know they are developers. But there are those that are also more concerned with security. They want to ensure that the code they're writing is actually very, very secured. So you need to look into the language you're using to ensure that you're using this security best practices for your language. For example, if you're using something like let's say python, you want

to look into the security best practices for Python. And I use this as an example because you know, even simple websites I mentioned big hacks earlier

35:00
involving big organizations, but come on people, their websites have been hacked. If you if you want to take do some experiment, if you have a website, just go in there and check your logs, you know frequently to see what is going on or even put, just put a packet sniffer to see what is going on. And I bet you, you're going to be scared to see the amount of traffic coming at your website. So if you have to ensure that the code you're using, the language you're using to write your code is also also properly secured. So that your website or whatever it is you're building can be strong from security perspective. Alright, so just two things you could do from the code, just making sure that your developers are well aware of these things. And, you know, Charles also talks about shifting security left. So there's this concept of shifting your security left or right, what that means is that the more you shifted to the left, the easier is going to be for for you to fix. And the cheaper is going to be for you to fix as well. Which means if you have security as a mindset, if you take into consideration security, right from the time that you are designing your application, then is going to be a lot easier to fish than if you leave it until some later time. So you can imagine, for example, if developers can discover the box in their code as the application has been tested, it's going to be a lot a lot cheaper to correct, then when the application is already live. If the application is already live, then, you know, maybe you have to let your customers know, just like my bank, let me know earlier that I told you guys about. So ensuring that your developers, they start with security first, security first is the number one important thing to do here. Alright, then we can move from there to the container environment. In the container environment, you have to deploy your code to the container. Before you do that there's a concept of packaging your code, you have to package the code into an image, in this case, a Docker image, once a package that code into a Docker image, you have to upload the code to a registry. In this case, we can say a Docker Hub, which is a public registry. Alright, so it is your code board before you can run it, it has to be recited somewhere in some repository. Now, this repository could be public, or it could be private. Of course, from security perspective, you want to ensure that is private so that people don't have access to it. But I want to let you know that the battleground these days is on the container image. Because if I as a hacker managed to go into your image and inject some code, okay, we can be any cross script

37:49
code to make sure that when a user has click on that, it goes ahead and do some other stuff on some other website, or SQL injection, or whatever it is, if someone managed to pull that into your code, then you have a trade wait into

38:05
to manifest, it's more like a time bomb, just a matter of fact, a matter of time, and the bomb is going to explode. So we have to ensure that the registry where we store our code properly secured. And that is why most registries these days are going to have a way to scan the image, they're going to post canon into their CI CD pipeline so that at any point in time, they ensure that the image is actually safe, and it's still safe is save at the beginning. But it's still safe, even as they're using it continuously. So that if there's anything that they have to be aware of from security perspective, that they can take care of that immediately. Okay, so making sure that our registry is secured, making sure that our images are

secured is is very important. Okay, so within the good tonight yourself, you can also do a lot of other stuff. In security, generally, we want to make sure that you reduce your attack surface, okay, the concept meaning that and just mention it earlier as well, when people are trying to attack you is always at the point of entry. Okay, most security happen at the point of entry or the point of exit, because that is where people can come in, it's like you have your building, right? The easiest way for people to come in and attack you or steal or whatever is through the door, okay, because that is an exit or to the window because as an exit is going to be more difficult for them to come into the wall because that means they have to, you know, try to break the wall or something. So it's the same concept in security. Here. We need to ensure that we reduce our attack surface to make it very, very small. And some of the things you could do in that regard is to ensure that first of all the image that you're using as the base image, okay, if you go to next slide, please, I think is probably the last slide.

39:58
Last one

40:01
Next one.

40:03
Okay, here, thank you. So here, this is a simple file called the Docker file. If you wanted to build your image, this is how a Docker file looks like, you have to say from that from means you have to specify a base image, the image you are going to build upon in Docker in Docker environment, you have to the building layers, so someone has to build some image for you to build upon. So you always have to have that statement from so that the from image that is there, you have to ensure that that image is also secured. And you have to make sure that that image is properly hardened from security perspective.

40:45
For example, if you don't need to have SSH running, don't have SSH running on that server. Any application, you don't want to have running, remove it. So you can reduce your attack surface. If you don't want anybody to be able to change permissions, remove changing of permissions on that code. So basically,

41:04
any port, you don't need any application, you don't need any library donate need to make sure that all that is removed from the base image and also from your own application. By doing that, you are reducing your attack surface. So that's another thing you could do. Also, Docker containers in kind of they are tricky because they need certain privileges to be able to work to be able to run, which means in Linux, that could be running as root with root privilege. But that is dangerous, because anybody that has access to that image could as they could a lot of things as the root user. So it's also good to be able to reduce the privileges of the dock a user that is trying to run the application in your image. So things like that you have to be able to take care of a whole you know, series of things within the container environment, they are they are trying to use for your application. Can you go please?

42:10
Back to where it was.

42:14
All right, thanks. So that is the second layer, which is the container containerized environment, then we need to move to the cluster environment. This is the Kubernetes cluster. Now, as the orchestrator. This is what people cannot do without these days when they are running their micro services based applications. And Kubernetes itself is very complex, he has a lot of moving parts. And part of that I'm going to show you some of them if you go to the next slide, please.

42:45
Okay, so here's what a Kubernetes architecture looks like, just to be able to run your micro service application, you have to deploy Kubernetes, in what we call the cluster, the Kubernetes cluster. And this comprises of a lot of services. Kubernetes generally is divided into two layers, you have the control plane, like you can see here, the control plane to the left of the screen, and you have the nodes to the right of the screen. So inside the control plane, you have the Kubernetes API server, you have the cube scheduler, we have the Cube controller manager, they have the Cube controller, and then they have a cloud controller manager of the Cube controller manager, then sitting right here is our key value data store. This is where all the information about our cluster, this is where we're going to get installed. So right here, on this cluster here, we need to make sure that certain things here are secured. Okay, we need to make sure that our cube API server is properly secured, we need to make sure that our ad CD server is properly secured. And surprisingly, how secure are those services, these two out of the box depends on the tool that you use to deploy your cluster. If you deploy the cluster yourself,

44:06
or you're not very conversant with the Kubernetes architecture, then you're most likely going to leave a lot of holes for people to come in. And also, even if you use a deployment tool, which often that is what people do. Some of these deployment tools, they are more secure than the others, they think more deeply about security, security than others. So

44:31
if you have a deployment, too, that kind of doesn't give you the proper security for any of these two services or components of the controller node, that could also expose you to a lot of tense. So let's talk about the cube API server. The Cube API server here is very, very important because this is the entry point to your cluster. Just like I mentioned earlier, this is the entry point. This is where everybody comes in true. And so for people

45:00
Come in here, if they happen to maybe login with the wrong credentials, then of course, they may be able to go into your cluster, and then escalate their privileges to be able to do most of within your environment. So one of the things that you have to make sure you do, especially in the Kubernetes cluster is to ensure that, at a minimum, this is going to be exposed on HTTPS endpoints. So that make sure that your certificates are properly configured to enable, you know, to secure that environment. Okay. So also making sure that you even restrict access to it. Because the way that Kubernetes

architecture is designed, you don't really have access, you don't really need access to the API server from outside the cluster, not even from the internet. So the common thing to do is to just make sure that that is blocked from the internet so that no one can have access to it. And then the administrators have to be also be careful when they're when they are trying to interact with it, because anybody can, you know,

46:03
man in the middle, try to see what is going on there, maybe intercept the traffic and do whatever they need to do with it. But yes, Kubernetes API server is one thing to really pay attention to. And then the second one here is your ad CD server. Like I said, this is your key value, key value store data store. And so everything your cluster, including Kubernetes has something called The Secret secrets, and config maps, they are all stored within this environment. So literally, if someone have access to that box, that's all they need. That is the good, that is the asset that they're going to be going after. So protecting that asset is also very, very critical. And often what you are going to see in architectures that are properly secured is that no one can talk to this sed server as said the cube API server. And in fact, even though it's the only the cube API server that can talk to it, you have to ensure that there's mutual TLS, that this has to authenticate with this, and this has to, you know, authentic and bark before that communication, we can be established. Alright, so we have to make sure that these two are properly secure from that perspective. Then when we come to the node, the worker nodes here, you see the architecture of each one of these worker nodes are the same, you have the cubelet, then you have the cube proxy, the cubelet is the agent of the API server on the node. So instantly, you can tell that on every node, the cubelet also have to be properly secured, because if you don't, then either someone is able to break in here, and then have access to your workload. In this case, your workload is your your containers, which are running inside

47:48
that node. So once they have access to those, your workload, then of course, they can do any, any sort of half of that they want to do, your application can be compromised, they can from there try to you noticed your data, they can from their jump to other stuff to do some escalation of the of the

48:07
whatever triad it is that they're trying to do. So that has to be properly secured as well on each one of the nodes. Okay, so from the architecture perspective of Kubernetes, you can see that

48:21
there are a few things which you need to make sure that you take care of otherwise, you could be in trouble. Remember, our Docker is here. So you want to take care of the Docker runtime, which is the container aspect of the infrastructure I'm talking about, then the next thing is to move to the Kubernetes architecture. And then to make sure that your Kubernetes architecture is properly secured. And just to give you a better knowledge of what a Kubernetes cluster looks like, for those of you who may not be familiar with, I'm going to just quickly share my screen for a second to show you a Kubernetes cluster. So I'm just going to share here with this one.

49:00

And share that. Okay, so I am already here on my, my cluster, as you can see. And when I run a command that command is is academic cluster. So in the Kubernetes environment, anything you want to do, you use a command called coop CTL, commonly called coop cuddle. So coop cuddle is your interface to the Kubernetes API. Any command I run at this point is going to interface with the API server, the API server doesn't respond, then of course, it means it is down or something like that. So here I'm going to see my number of nodes. I can say coop cuddle, get nodes. And

49:41
it brings that I have four nodes. The master node is there. And then we have three worker nodes. If I want to see more details, I can see my notes Oh wide, and that will show me a lot of information about my notes. So I know the IP addresses, they're running on stuff like that. So when

50:00
deploy a cluster live is

50:04
comprising of these four nodes, there are basically three steps that happens before you can actually get access to this to this response. So what from the time I type this command, to the time that there's a response, three things happen. First is going to be the authentication. Okay, I have to get authenticated, you don't see that happen here, because I'm using certificates. So Kubernetes is designed to have certificates so that at the point of entry, we make sure that at least that is taking well, well taken care of. Alright, so after the authentication, there's authorization, and Kubernetes can easily use what we call the airpark role based access control. So if I did not, if I create a user here, and I don't want that user to be able to do anything other than list my pods can do coop cuddle get pods.

50:57
Post, this shows me my application that is running. So here it shows I have some application running here, see all of them says running, if and then wanting this to be able to see the pods, that is the macro services that are running, take this to be the micro services that are running. If I if I want someone to be able to only see the pots, you could not create them, I could create a role based access control and give that person just the ability to list the pots. Okay, so in my Auerbach, there are a lot of things I could do. Then the third thing that happened is what we call the admission controllers. So again, you can see three A's authentication, authorization, and then admission controllers in the admission controllers, I have to make sure that if there's certain policy that I do not want to get executed on my cluster, I can block it. For example, a common example is if I do not want images to come from certain repository, maybe because that repository is considered to be public, and it's not secured enough, I can block it at my admission controller level. So Kubernetes itself is built with lots of securities. But if you're not able to apply those security practices very well, that will come back to haunt you in a very big way. Okay, so I'm going to leave that for now. Just giving you an idea of what a Kubernetes cluster looks like, some of the few things that you could do right there. So I'm gonna stop sharing so I can go back to where we were before.

52:34
Stop share.

52:38
Okay.

52:42
Yeah, I'm gonna try and round are busy. Okay, yep, here we go.

52:50
Okay, so yep, this is one of Justin in the architecture. Okay. So in the within the Kubernetes, there's still a lot of things to do our workload, C have its own aspect of security, okay, that we can take care of, for example, can do something, it's got a PSP, Paul, pod security policies, you can do things like network policies. So there's a whole lot of stuff we can do within this Kubernetes environment to make sure that everything gets secured, then I'm going to go to the final C, if you go back up, please. The final C of our security oneone. The final C here is our cloud environment. Okay, in the cloud environment, you have the option to deploy your own cluster using any of these using any of these tools I just talked about, or that I mentioned, or you can actually use a manage Kubernetes environment. For example, on AWS, you can use something called Eks, elastic Kubernetes service, or you can deploy your own cluster. In any case, when you use someone else's environment, you need to ensure that you apply with the security best practices for that platform. In AWS, we have this concept of the shared responsibility model. In other words, there are certain things that AWS would do as a cloud provider. But there are certain things you the user have to do as a user, if you don't do your own part by AWS does he is you are say going to be in trouble because you haven't taken care of certain things you need to take care of. For example, if I don't want if I create my cluster, and I do not want certain posts to be exposed, I need to block them using security groups. If I don't block them using the critic groups, I could easily get hacked. If I quit my EC two instances, for example, as well. I need to make sure that maybe those images follow best practices of making sure that I hadn't them and only put things that I want to be in there. And I can use things like the CIS benchmark which

55:00
is like a checklist of 1000s of things I need to make sure I do for ensure that my cluster is up and running in,

55:08
in a secure manner. So if I don't take care of those kind of things for myself, also, I'll be I'll be I could easily be hacked. So the principle of shared responsibility says that, while the cloud provider does do what they need to do, you also have to make sure you do what you need to do. So that will be my final See, that I put here. And I would like to round up saying that, you know, Kubernetes, generally, is where people are deploying all these applications these days. And so that's a common adage that where the eyeballs goes, that is where the hackers go. And so with this, since microservices are deployed mostly

55:46
predominantly in their in a Kubernetes based environment, it means that when we're deploying our micro services, we need to make sure that to take care of for these four, four layers of see that I tried to highlight here. Okay, so with that, Charles, I think I'll pass it back to you to see what else you want to

say before your random. Okay, yeah, thanks. That's it. I mean, I think that was really helpful. Took a bird's eye view and onion, kind of substance architecture, and you're really made sense. And I liked how it kind of simplified the way of understanding the various architectures, I think it was really helpful. So I'm just gonna jump to the last slide. I know we're running a little bit over time. But I wanted to ask you definitely me being that you've been in this industry for a while, you've seen a lot of this, you know, a lot of the cloud development

56:41
scenarios play out real time with companies and you actually teach classes. If someone's wanting to learn more about this information. What do you suggest as the first step, you know, to get get exposed to some of the topics that we talked about today? All right, yeah, definitely. I mean, you and I, we run courses

57:04
on security, and cloud, native security, Kubernetes, security, and all this kind of different trainings that we do, both online and in classroom based training. So I would definitely suggest that you check those out. And, you know, attend. The thing with training a lot of time is it saves time, because someone who has been there is able to put together a content that will save you tons of time. So yeah, that will be my first recommendation. And then also, it's also good to go out to meetups and to, you know, network with your colleagues and hear what they are doing and see what you can learn from them. So we also have a meetup here called a Kubernetes. And cloud native, not Dallas, we meet like the third Wednesday of every month. We're close to 2000 members now. So one of the largest meetup groups in Dallas area. So yeah, I mean, you guys feel free to also register online and join us these days is all online. And also webinars like this, like chance mentioned in the in the beginning, we intend to do a lot of this stuff. Because I mean, awareness is huge. Letting people know about this, since it's huge, and individually as become aware of all these challenges, things will start to take steps to make us

58:25
better secured in our own environment, even if it's just making sure that we don't put our password on our screen on a laptop screen, or even making sure that we will set MFA is for our accounts, because they may face can be very useful as well. So yeah, I mean, these are some of the recommendations that I would give to our members.

58:47
All right, great. Thanks for that feedback. Excellent. Excellent.

58:52
information there. Dr. Demian. I think hopefully, the information I shared earlier on the high level aspects and culture around dev SEC ops, that was helpful. And we look forward to running another session. And we will advertise this out. And hopefully, those that join will join again, and we look forward to having you have you have any questions or feedback, we'll send out a link to this, this recording. And yeah, we'd love to have any feedback on this or any topics that you might want to explore further sessions. So with that, we're going to close the session and thank you very much. Have a great day. Bye bye. Thank you and bye