

Interviewer and Dave (employee for one of the “Big Four” banks), “I” and “D” respectively
(Transcription starts)

I: “In all seriousness, how have you been?”

D: “Yeah, pretty good.”

I: “Good to hear, good to hear. Got any nice elixirs, for this interview of ours?”

D: “Alexa?”

I: “Uh yum, yum, drink? Soothes, the throat? ”

D: “Oh, right, Elixir! No. I had a coffee this morning, that's it.”

I: “My corner store had a sarsaparilla Bundaberg, so I grab that on the way out”

D: “Huh”

I: “I always forget what sarsaparilla taste like, I buy it-“

D: “Awful.”

I: “-and then I think it tastes terrible and don't buy it for two years, so let's have a go”

I: (*Schluupr*)

I: “mouthwash! I can see why they had it in the discount bin, tastes like medicine!”

D: “Lesson learned?”

I: “For a few more years, at least. Anyway, shall we begin?”

I: ”Dave, thank you very much for taking the time to do this little interview of ours. Not to waste time I'll just jump right into the questions. First, would you please tell us about your IT work: what exactly do you do? ”

D: “What exactly do I do? I am an automation engineer. I automate anything that needs automating. I work for the operation centre, so we look at all the systems throughout the bank, whenever something goes wrong we look at how to automate a system that will quickly fix it in the future, as well as different kinds of monitoring; we also do disaster monitoring, so looking at different kinds of natural disasters, looking at what sort of asset risks they have, “is it close to a branch?”, that kind of thing. It's a mixture, we are a small, agile team, able to jump on anything that needs jumping on.”

I: ”Interesting, interesting! So, automation engineer? Is that your official job title?”

D: ”That's the one.”

I: “So it seems like you are something of a jack of all trades, whenever there is a not-so-automated nail that needs automating they throw you at the job; are there any recent instances of you being given a conventional task that you needed to automate your way out of?”

D: Let's see, something unconventional... We were doing Oracle database automation."

I: "HMM."

D: "So automating recovery steps for switchover with the Oracle DataGuard system, it's nothing unusual. It's all pretty standard stuff."

I: "Fantastic stuff, so, please tell us about the industry you work in."

D: "That's pretty broad.

I: "Elaborate."

D: "Well, I guess you could say I work in two industries, do you want to hear about banking or engineering?"

I: "We might as well do a little bit of banking here, I don't think will touch it much later so, tell us about banking then!"

D: "Yeah, it's pretty good, everything is very slow moving, it is very risk averse, everything is done very cautiously, it's a bit slow, which, if you get really passionate about it and you just really want to make something happen, and you're really eager to achieve an outcome, that's somewhat counter-productive in this industry, I think you've gotta just roll with it, it happens when it happens."

D: "You have to have some level of passion to want to make things happen, but you can't go too hard, I think it's an industry where you need patience."

I: "Interesting. That's a very astute analysis, I'd say! What other kinds of work do you have to do?"

D: "I have been doing a lot of infrastructure work for the team. All of our automation efforts require servers, so I have previously had some DevOps experience so I have enjoyed automating our infrastructure, throwing it on terraform."

D: "I'm working on automated infrastructure testing at the moment, so we can daily test our infrastructure code, spin up a stack, do some technical verification testing with it, and then just destroy it instantly."

D: "I think that's the beauty of being in the cloud, you can just spin up infrastructure so easily, see if it's working, then get rid of it. So that's kind of fun."

D: "There are those kind of opportunities, when you are working with approved software, you can use more modern patterns, it's just if you want the latest tooling, or the latest version of software, that's where the banking slows it down, but any sort of techniques you want to do, you're welcome to do that."

I: "So big on the continuous integration then?"

(Audible, pained sigh)

D: "Ah, try to..."

I: "Perhaps big on continuous integration, not so much continuous deployment?"

D: "Yeah? I think everyone would like to be doing continuous deployment, but, you've gotta have a good record, you've got a show that you can be trusted with that."

D: "The automation we work on is fairly low risk: if something doesn't work, we still have the manual procedures to go in, it's just a little bit slower, but that's all factored for, so if our stuff goes down, it's not the end of the world, but for the sake of our team we try and keep it up."

D: "With the deployments we are trying to be risk averse, we don't mind doing it manually if it means we're gonna do a better job, hopefully with a lower risk of failure."

I: "Interesting! Seems like a very levelheaded, sensible approach, like if you've got a quarter of the country's economy, resting on your shoulders, seems like the right way to be going about things."

I: "But I am curious, you say you bring up servers, play around with them, see if they were okay, so rather than having all of these procedures in place that automatically integrate and test, you can, just, of your own volition, say "I wonder if this works" and create an instance, play around, see how it pans out, kind of like learning through trial and error"

D: "Yeah, yeah, for sure, if you want to deploy a new infrastructure, you can just spin up a test host, all of our infrastructure has to run on terraform, so if you want to see what a new feature looks like on a bit of software that you were using, like we use airflow for example, we can just throw up an airflow server on terraform, see what this new package looks like running on it, or see what it looks like when we make these Docker changes, make sure it all works, and then we can look at integrating it into other environments. We can pretty much have unlimited environments with terraform, which is great."

I: "Fantastic! So, you mentioned earlier a bit about trust, and I wasn't sure if you were talking on the individual contributor or team level, so who are all the different people you interact with in your work? Please tell us about them."

D: "So I've got a team, we've just had a new intern start the other day, there is him, we have three other engineers, myself included, and then the senior engineer on the team, and then we have our manager who we interact with daily, he is great and I'm so happy with the team in general."

I: "So a very close-knit team then!"

D: "Yeah, we all get on really well. I think that's a great culture at the bank, everyone gets along really well, and I can't overstate how much I enjoy working here because of that."

I: "Have you worked in larger teams before, did you think that was preferable or do you prefer the smaller teams?"

D: "I've worked in teams that were a little bit larger, but I think it's really nice when the team really gets along, like a real good culture, sometimes you get someone on the team has a little less sociable in the workplace, they just want to "do their job, log off" kind

of thing, but it's nice when everyone is really friendly, happy to go for a drink afterwards, you'd happily hang out with them outside the workplace."

I: "Nice and close-knit, then, fantastic!"

I: "Tell us about your other interactions with IT professionals outside your main work group."

D: "God, they're vague."

I: "Mysterious contractors then?"

D: "I guess outside my main work group, it's quite a large organisation, so if I need to do something, like, if I need a firewall request put through, or I need some service account created, or a role provisions, then there's lots of teams, which are a good distance away from me, that I need to interact with, and if it's just a one off interaction, then it's generally pretty formal, and rigid, and they have an engagement process they want you to go through, and you're just going through a set of steps that are written down, it tends to be pretty inflexible, some of them will get a bit sick of it, it's gonna be pretty painful for them..."

D: "But if you need to file a more than one request in the space of a couple months, and you start to recognise each other's names, and then, sometimes you take it outside the request process, you start messaging each other, you manage to build up a bit of rapport with them and then the whole process becomes just a bit less rigid. That's nice, and it's good working on a project as well, sometimes you'll get a direct engagement with someone from the firewall or service accounts team, then you get a real rapport with them and it's great to be able to just reach out, to have that contact in mind, to be able to go "ah, I know who can help with this", like if I'm wondering if the firewall is working this specific way, or "what's this environmental difference", instead of filing a request that sits in a queue, that I'll get back two weeks later; to be able to just reach out to them... but I think that's like a scale thing, with smaller companies, I'm sure you've got that more intimate relationship with everyone, comes a bit more naturally, but anyway."

I: "So, tell us about your interactions with clients or investors, or, I suppose, if you work, primarily internally, the other teams or employees that are the clients for the products you build."

I: "I'm just thinking in terms of, if you're building tools by other employees, then, in that sense, the other employees are your clients."

D: "Yes, essentially. They've been great, I think I'm really lucky to have the people we've been working with, they have pretty clearly defined requirements, they've been incredibly responsive, it's been great dealing with clients who show you exactly what they need, you go out and build it, come back, and they say "Oh it's fantastic"; any adjustments have had have been minor, I imagine it could be very tedious working with certain clients, but it's all been a very positive experience for me."

I: "Well, that sounds like an incredibly satisfying work experience!"

D: "Yeah, I can't talk highly enough about this company, I'm loving it."

I: "Fantastic, fantastic, I'll make sure to let everyone know to try and get a job at [REDACTED] then."

I: "What aspects of your work do you spend the most time on, please tell us about this."

D: "Debugging."

I: "Care to elaborate?"

D: "No."

D: "Jokes aside, when I'm building some sort of greenfield tech, where I'm building something from scratch, which I really enjoy it, but I think the majority of it is, well, coming up with a big picture isn't too hard, "this is the system, this is what it should look like", and then you just go and write it all, and then it's all built, building it isn't exactly the hard part, ironing it all out, it gets harder when you're doing all of your testing, you turn on the server, one morning, some sort of patch has gone through, all these little changes, causes some issue, and fixing it, all these little bugs, working out all these different scenarios, I think building the initial thing isn't the hard part, but reinforcing it, getting it production ready and working through bugs, that's most of my day."

I: "To come up with an ridiculous analogy on the spot, it takes one afternoon to reassemble your car's engine, it takes three months to get it turning over again."

D: "Yes, exactly."

I: "I somewhat feel that's the one line, universal answer for any software engineer asked that question, but, oh well."

D: "Yeah"

I: "Perhaps some things don't go without saying, nonetheless."

I: "So, what aspects of your work do you find the most challenging? Slight nuance here, but I think there's a significant enough nuance between the two questions."

D: "Where do I find the most challenging? Paying attention."

(Both laugh)

I: "So..."

D: "Nah, in all seriousness, I guess navigating the enterprise."

(Neither the interviewer nor interviewee are big enough nerds to have jumped on the obvious opportunity for a star trek joke here. They must both be #FakeNerdGirls)

I: "Well, you said it's a rather large business, the scope of which is beyond human comprehension, the sort of thing that might drive a man, deep into madness, so trying to work out why Marvin in accounting is responsible for the AWS credentials, or whatever..."

D: "Yeah, I guess so, it's very dynamic, there's a lot of things going on, I guess some clients aren't as good as others, changing requirements and whatnot, thinking of other issues like, what is the most challenging part? "

D: "I guess the security is one part of things, building everything to security requirements, sometimes it's understandable, but it's frustrating, the kind of workarounds you have to do to hit some requirement and some of the requirements you can see why they're there, but they're just not practical."

I: "So all of the, if I understand, correctly, security, certification, and requirements that you need to fulfil, making sure that you're ticking all the boxes, crossing all the "T"s, dotting all the "I"s..."

D: "Sometimes, there's like a really good way of doing things, but security is not happy."

I: "So you have to do something in a much more difficult way in order to fulfil security requirements?"

D: "Yeah, I don't know, it's like 99% of the time you feel like you could've been done so much earlier, but you're adding in all this extra stuff in for security requirements, but you get why they're there, a lot of them are very logical, and then some of them are like "come on man, I get where you're coming from, but, you know, you're not living in reality, here"."

I: "I imagine most of these rules are written in virtual blood, and at some point, someone who was not living in reality, made a very horrible mistake, but... out of curiosity, do you think any of the security paradigms have become outdated with the way system architectures have changed, or do you think there might be someone stupid enough to make that mistake again?"

I: "Or do I misunderstand? Is it no so much "you're not living in reality", in so that "this is not applicable", is it just that it is completely impracticable to do these sorts of things in a given context?"

D: "Well, it's more like "We need this software to do this", "Well, it was not written to do that.""

I: "Ah."

D: "That is not a feature of the software."

D: "And it's like "we need it to happen anyway", and so we end up jumping through all of these hoops to fit some requirement onto some bit of software, and it's like, there's extenuating circumstances where I don't think the requirements are entirely applicable, where in the worst case scenario isn't even that bad, if bad at all."

D: "One such example was when dealing with public data. For instance, dealing with branch locations."

D: "You wouldn't exactly think public data needs to be secured all that much? We still have to jump through all of these hoops to protect the security and privacy of data that is available on our website for the entire world to see."

D: "If you ever thought it was difficult to find the opening hours of your local branch on our website, if you want to access that information from the inside, you need two engineers on opposite sides of the building, to turn a pair of keys simultaneously, while both entering a cryptographically signed passcode from the president of the United States into a teletype while blindfold"

I: "Quite an example, I feel where you're coming from, but just to make sure I understand the core of your concerns correctly, the security requirements were never part of the original scope in some cases, and, having to retroactively, add them into a mature code base is a, painful experience to say the least, if I understand correctly?"

D: "Yeah, so your engineering because you want to build something, you want to deliver value for the company, you want to make something that's useful, that people are gonna use, so you work very hard, and you build this thing, not even sure how it's going to take, how people are going to react to it when it's built; they have these requirements, but you just make it, you just see if it's going to work, if people like it, and then you have this thing, people love it, and the powers that be say "Great! Let's move it into production!", and then it's "Okay, let's get security on the phone!" s'like "Oh god here we go.""

(Interviewer wheezing with laughter)

I: "Finally, so it seems like an awful lot of satisfaction from creating value for the company, creating value for the customers, do you think you could share an example of the work you do the best captures the IT industry?"

I: "(Without breaking too many NDA's.)"

D: "Captures the essence of the IT industry..."

I: "The essence of the IT industry!"

D: "Being? What's your take on what they mean by that?"

I: You see, I thought I was the one asking the questions, doing the interview, but here I am, it's turned around on me."

D: "No, I just mean clarify question."

I: "Yeah, I guess that is the interview is job, isn't it? The essence of the IT industry, I guess I could throw around synonyms like zeitgeist, but just to make things fun, I think it would be a project or a task, something that is a part of your job that represents the current state of the IT industry as it is."

I: If you told a story, it would be very easy for someone to say that sounds like that's the sort of thing that's going on in the early 2020s, cloud starting to mature but there's still people with mainframes, it's a very "Google hasn't quite taken over the world yet but at the same time, they're getting pretty strong", when you hear a companies acquired another, you're not excited any more. You're just kind of gutted, and here I am trying to capture the essence of the industry. I don't know, I guess I'll have to leave it up to your interpretation."

D: "What am I doing?"

I: "If it was the mid 2000s, I'd say, the IT crowd is the absolute answer to this question, but, it's been a fair few decades, and a fair few cultural shift from that being even remotely appropriate any more."

D: "Alright, I guess this infrastructure testing thing that I'm working on at the moment is a pretty good snapshot of where we are"

I: "Elaborate"

D: "Now that we're on the cloud, we can do things we never could, spin up a server for an hour, run a set of tests on it, and then kill it, I guess that is really "wow, we are really far advanced, this is some really cool stuff", and then, on the other hand, I'm using shell scripts to do the testing, you know?"

D: "We're at the point where we've got this very cool technology, but it's not like so mature that you're doing it all in YAML. I feel like the fact that I've written half of it in YAML, and the other half in shell script, is a good signifier of where we are, of where the IT industry is. We are shifting into this whole codeless, defined state, kind of system, we're moving away from taping things together with code, we are moving towards a more lockdown paradigm for just achieving a single task.

D: "Terraform is not something you fizzlebuzz in, it is something you define a state in, but I feel like we're not all the way there. We could do lower-level infrastructure testing, like to test if a whole system is up and running, that still happening in a shell script. I feel like we're in a transition period, that sort of captures the state of where we are."

I: "So, the magical powers of instantiate an ephemeral 1970s mainframe, the sort of stuff, Dennis Ritchie couldn't have dreamed of, still with bits of bells labs rattling around inside."

I: "The transitioning away from that, perhaps into something more functional?"

I: "You mentioned YAML and describing like..."

D: "The half that's in the future is the YAML, and the spinning up infrastructure, and the half that's in the past is the shell script I'm using. The future is that it's all in the YAML"

I: "What's the, what's the word that begins with D, that describes, Declarative!"

D: "Yeah"

I: "Moving to a more declarative system, I do wonder, even if the servers are virtual, and even if they are ephemeral, they still live on cruddy operating systems with lots of shell scripts, but it seems that we're almost halfway transitioned away from that from what you've deceived, that maybe one day all of that will all be gone? And it will just be the declarative-"

D: "Yeah, I reckon, we'll work out something better."

I: "I wonder if you've got any inklings what it is? If the way you're doing things now, the Industry will just iterate on that, or if you think there's an all new revolutionary paradigm that will pop out of the ashes, or-"

D: "You know, I think I'm too junior to have a proper opinion on this, I don't think my understanding of software principles are sufficiently mature enough to make a proper prediction on that; I think we will move away from these... (pause), I feel like what we do restricts, like restrictive practices, but in a good way, I think things are going to get a lot more siloed, and those are going to enable things, like I think self-healing is a big one, you get it with defining state if we try putting it into a state, and that doesn't happen, then there are ways to handle that sort of thing. That's what these paradigms are going to bring and enforce. That's why I think that'll be the future, instead of just "you write a script to achieve an outcome, it s**ts its pants, ok what now?"

I: "So the secure safety of a comforting hug rather than being stuck somewhere unpleasant?"

D: "Yeh."

I: "There's a few tangents I could grab onto there, but all in all I'd say that's a fantastic point to end on."

D: "At exactly half an hour."

I: Yes. I certainly... it's funny, I just looked at my transcriber, it says 29 minutes, I thought we're probably coming up on time, but no, in the call, there, 30 minutes; regardless, I... Don't think I can come up with any further questions that justify your intellect."

D: "No."

D: "No, not that you can't come up with any questions, But no, "my times with all that much." Like...

[redacted comment on remuneration]

I: "That certainly not going in, anyway, thank you very much Dave, from BIG4 bank, it's been a pleasure speaking to you, I'll make sure I'll email you through a transcript of this interview just as soon as my super duper super computer is done, transcribing it all."

I: "Not like I now realise I'm going to have to spend four hours, transcribing it by hand."

[redacted at request of interviewee]

I: "I'll let you get back to that fun automating you love so much, here's hoping the security team doesn't give you too much of a hard time."

D: "As long as I don't have to deal with the science team, any days a good day"

I: "Cheers man."

D: "See ya."