YouTube Summarizer

Transcript:

I used to feel overwhelmed with being a cloud engineer, but after working for a large bank in London for over three years, I've learnt some important lessons that has fast track my career. In this video, I'll share the five key lessons I've learnt from the past three years, including my single biggest regret that held my career back many years. Basically, this is the advice that I wish I knew three years ago, starting with the first key lesson. Before I got my job, I spent a lot of time studying for certifications and reading documentation around AWS. And I don't completely regret this. After all, it probably did help me get a job. But looking back, I don't think this was the best thing to do. Truth is, certifications provide a great foundation, but they don't replace real world experience. Imagine two engineers interviewing for the same role. One has four AWS certifications, but has never built anything. The other has no certifications, but has created three really impressive projects. Who do you think the hiring manager will be more excited to talk to? In my experience, definitely the one with real projects. And this is a mistake that I made too. When I was trying to get a job, I was too focused on certifications and theory. If I had to estimate, I'd say 90% of my study time was spent on certifications and only 10% on building real projects. But if I could go back, I'd change this and aim for around 80% of my time spent on projects and 20% spent on certifications. Hiring managers want to hear about the time you use DynamoDB streams to process real-time data. Or the clever way you reduce cost by 30% using spot instances. These are stories that will make you memorable and prove you can do the job. But does this mean certifications are worthless? Well, no. They're still good for learning best practices and proving your dedication to improving your skills. And many job listings still prefer candidates with specific certifications. For me, certifications have actually become more useful after getting my job. When I started working, I was getting practical experience every day. So I didn't need to build lots of stuff in my spare time, as I was already getting this experience through my job. But there were definitely still gaps in my knowledge, especially around services that I don't use regularly. In this case, certifications became really useful for me because it filled in these knowledge gaps. So no, certifications aren't completely worthless. But if I could give my past self some advice, it would be to definitely prioritize building things when first starting to learn cloud. But there's actually something even more important that I wish I did. And this is nothing to do with coding, cloud, or even technology. So imagine there's this guy, John, who just started a new job as a cloud engineer. He's been given a task to do, but he's completely lost. The thing is, instead of trying to figure things out on his own, John keeps going to his co-worker Sarah to ask for help. Hey Sarah, I've run into this error, can you help? Hey Sarah, I don't know how to do this task. Can you just do it for me? Now, Sarah's trying to be helpful, and does help John, but naturally, she's getting a bit frustrated. It's like, come on John, at least try to Google it first or search the documentation. Now, don't get me wrong, there's nothing wrong with asking for help. In fact, sometimes it's the only way to find out the answer to something. But an important lesson I've learned is that there is a right way and a wrong way to do it. So what's the right way? I found that it all comes down to paying in some effort yourself. Let's say you're trying to build something on AWS and run into a bug. Before you go asking the senior engineer, take some time to Google it and troubleshoot yourself first. It's also worthwhile searching through teams or Slack channels to see if anyone else has had a similar issue. But okay, what if you've tried all of that and can't find anything? Well, when you do ask for help, you can't just be like, it's not working, I don't know why. You have to give them some context. Explain what you've tried and what specifically you're having trouble with. I've realized that the more specific I am, the easier it is for the person to help me. For example, a request like this might be quite good. I've checked the security group settings on the EC2 instance and verified the VPC configuration as per the internal documentation. I'm still facing connectivity issues. Any ideas on what I might be missing? So the real lesson here is when you respect four of your colleagues' time and you show that you're trying your best, they're way more likely to want to help you. But there's something that I wish I did earlier that would have actually decreased the amount of questions I had to ask. When I first started my job, I remember an instant I was involved in where one of our applications had some connectivity issues. Users were reporting that they couldn't access the application at random times, and it was causing a lot of frustration. At first, stupidly, I thought it would be an easy fix. I checked the application logs and then it's seeing obvious errors. I looked at the EC2 instances running the application and they all seemed to be healthy. As I was going down all of these rabbit holes and dead ends, a senior engineer spotted the issue quite quickly. It was to do with how the networking was configured for the application. This was a bit of a wake-up call. I realized that my networking knowledge wasn't very good and that I had neglected to study it properly. I couldn't rely on my certifications or just try to learn cloud and be a good cloud engineer. But what impacted this have on my career? Well, from that point, I made it a priority to go back and make sure I had a solid understanding of networking fundamentals. Things like understanding exactly how data travels on you load a web page, how networks are structured, and basic network security. If I had focused on learning this sooner, I would have been able to troubleshoot issues much faster. I also would have taken on more difficult projects which I probably shied away from because I wasn't confident enough in my networking skills. It's really one of those things that you can't hide from as a cloud engineer. So better start building the skills as soon as possible. During my first week, I remember sitting down with my team and discussing their infrastructure setup. Every piece of infrastructure from servers to databases to network configurations were all defined through Terraform code. And this made me realize just how important infrastructure as code was. Now, I had heard of Terraform before and did some basic tutorials, but I definitely wasn't confident enough to start managing all of this complex infrastructure. So I spent the first few weeks on my job feeling a bit scared of breaking something when I was asked to make changes to the infrastructure code. You know, the thing is Terraform wasn't something that was covered in any of my certification study. I had learned about cloud concept sure, but I didn't have any experience with the actual practical day-to-day tools. And this is something I see quite often with other cloud learners too. For example, a lot of AWS labs online involve setting things up manually in the console. But this is unrealistic in a real job. You'll probably manage all of your infrastructure through code. So what's the lesson? Looking back, I could have saved lots of stress and confusion if I had prioritized learning Terraform earlier on. I think learning Terraform is just as important as learning about cloud services like EC2, especially if you're trying to get a job. But out of everything I've spoken about so far, I think the next lesson is the most important one. When I first started my journey in cloud, I was pretty much only focused on learning the technical skills. I ended up neglecting one of the most important aspects of career growth. But looking back, I realized now I was missing out on a huge opportunity. So I remember when I first heard about these different AWS meetups in my city. I was a bit hesitant to go because I thought I didn't know enough yet. I was scared that someone experienced would ask me a really hard technical question and I'll be exposed as a bit of an imposter. But when I finally started attending these events, it was the complete opposite of what I expected. The people were very welcoming and eager to share their knowledge. In fact, I noticed that most of the time, people were more interested in talking about their personal lives than about the technical stuff. That experience completely changed my perspective on attending these events. I realized it's not about being the smartest person in the room. It's about being open and building relationships. And from then on, I tried to attend as many of these events as I could. I actually regret not attending these events even sooner. But why has it had such a positive impact on my career? Of course, I've gained a lot of technical knowledge and improved my problem solving skills. But it's the relationships that I've built that have been the most valuable thing. You've probably heard the saying, it's not about what you know, it's about who you know. And this is true. I'll call into a hub spot around 85% of jobs now filled through networking. But aside from this, I think it's great to meet people in the industry just for the sake of connecting and sharing experiences. It's great to have other people around you that understand the challenges that you're going through and where everyone can support each other. When I was starting out, I think I underestimated how valuable this is. So, where would I recommend finding these events? In the UK, I think MeetUp.com is a good platform. And this is actually what I personally use. Going to these events is quite fun. And this is one of the positive things about being a cloud engineer. And at this point, you might be thinking that cloud engineering is a good career choice for you. But let me tell you, it's not for everyone. There are definitely negative factors too that you might not be aware of. And you need to be okay with these. Otherwise, you may end up wasting a lot of time and money trying to get a cloud job. So watch this video to find out the reasons why you shouldn't be a cloud engineer.

Summary

I used to feel overwhelmed with being a cloud engineer, but after working for a large bank in London for over three years, I've learnt some important lessons that has fast track my career . Before I got my job, I spent a lot of time studying for certifications and reading documentation around AWS . But looking back, I don't think this was the best thing to do . I was too focused on certifications, but they don't replace real world experience . Hiring managers want to hear about the time you use DynamoDB streams to process real-time data or reduce cost by 30% using spot instances .