

## YouTube Summarizer

### Transcript:

What if watching more tutorials is the exact thing preventing you from getting a cloud job? I know it sounds counter-intuitive, but as a self-taught cloud engineer who now works at a top-bank in London, I discovered this the hard way. So in this video, I'm going to show you the exact system I use to escape tutorial hell, and you can start using this today even if you're an AWS beginner. But first, what actually is tutorial hell? And why is it so bad? Does this sound familiar? You find a tutorial, you watch it, you follow all the steps, and then you immediately start looking for the next one, and then the next one after that. But a few days later, you can't remember any of the skills you actually learned. Now, tutorials aren't inherently bad. They're great at opening your eyes to what is possible, but they're not a good way to build real world skills that you'll remember. Think about it like kind of learning a new language. You can watch video courses and read textbooks to learn the basics of French, but you'll never become fluent unless you immerse yourself until you're forced to have a conversation with a native speaker or reader French newspaper. When you're trying to land cloud, tutorials are these textbooks. I was stuck in this textbook phase two. I remember spending hours watching videos on AWS lander. I was trying to understand all the different settings and memorize the different limits, but at the end of the day, I still hadn't built a lander function that solves a real problem. I remember feeling like an imposter. The voice inside my head was telling me that I wasn't quite ready to build anything on my own yet, and that I had to watch another tutorial to, you know, really understand the technology. But looking back, this kind of mindset definitely helped me back. But what's the solution? Well, to explain now, we need to understand the principle of active recall versus passive review. Watching her tutorial is passive review. The information flows in one direction and it feels easy. But because you're not struggling to retrieve the information, your brain doesn't build strong enough connections to it. Building a project on the other hand forces you to use active recall. When you hit a roadblock, your forester stop and ask, okay, so how do I actually solve this? You have to actively search your memory or research the answer. That struggle is exactly what forges the strong lasting neural pathways. And this is crucial, because in a real cloud job, no one gives you a step-by-step tutorial. Your manager will give you a problem. They'll say, our application deployment is too slow. Figure out how to speed it up. Or we need a scalable backend for this new feature. Design and build it. Your job is to struggle, to research and to build a solution. And it's really important to get better at that skill. But what should you do next? How do you stop watching tutorials and actually apply this? Well, before we cover that, I'm currently building a step-by-step system designed to take you from zero experience to having an interview-ready portfolio of impressive projects. If this sounds good to you, join the free wait list at the link below and get early access with an exclusive discount. A lot of people get stuck at this point. They know they need to build, but they don't know where to start. So let's get specific and create a clear step-by-step action plan you can follow. Step one, build your foundation. This is for you if you're brand new to cloud. And I mean brand new. You don't know what EC2 is, what S3 is for, or what I am means. You haven't completed any courses, and you don't have a certification. You feel a bit overwhelmed, and that's completely normal. We will start here. At this stage, your goal is to get familiar with the basics. It's okay to do a course, and I'll recommend courses that act as training for the AWS cloud practitioner certification. Or if you want to stretch yourself, the AWS Solutions Architect Associate. There's plenty of these courses out there, including some free ones on YouTube. But whatever course you pick, here's the important thing. It should include a lot of hands-on labs. And with these labs, you have to challenge yourself. In my opinion, this is where you'll learn the most. When you get a lab that has a step-by-step guide, read the final objective first, understand the goal, and what the end result should be. Then, close the instructions and try to build it on your own. You should have a basic idea of what you need to do from the theory that the courses already taught you. But I'll be

honest, you will get a start. You will get error messages. And you will feel frustrated. I promise you, though, that feeling of struggle is the feeling of you actually learning. It's active recall in practice. Every error you solve is a lesson that will stick with you far longer than any video. So try to only watch the step-by-step instructions in a course when you are truly stuck. An optional but great way to support any theory that you're learning from a course or a YouTube video is to use a platform like AWS SkillBuilder. It's an interactive platform with hundreds of self-paced labs that let you practice on nearly every service imaginable. Oh, within a real AWS environment. The biggest benefit here, and the reason I recommend it for beginners, is that these labs run in a sandbox account. It's completely isolated from your own personal or business AWS account. This means that there's zero risk of you accidentally leaving a service running and getting a surprise bill for hundreds of dollars at the end of the month. And I think that safety aspect can be huge for your confidence. It lets you explore and click things without fear. But it is worth noting that to get full access to SkillBuilder, it comes out cost of about \$29 a month. So again, this is optional only if you want the additional labs. Overall, your goal in stage one is to build enough of a foundation to move on to stage two. And you'll know when you're ready when you're confident with the basic AWS services. A good measure for this is to take a look at the current curriculum of the cloud practitioner certification. If you're comfortable with the topics listed there, then you're ready to move to step two, which is to build your first real project. This is the stage for you if you already know the basic AWS services. Maybe you've completed a course for the solutions architect associate or you even have that certification. But your resume still has no projects on it. You have the theory but no real proof of your skills. If this is you, your action is simple. Stop watching new courses. I'm not saying you can't watch anything educational or theory based, but definitely don't start a new 50 hour mega course. Your new study time should shift much more towards building stuff. You should try to learn just enough to get past the next blocker in your project. But what should you actually build? A great starting point that I recommend for anyone at this stage is the cloud resume challenge. It's a project that challenges you to build and host your own personal portfolio website using AWS cloud services. But I think the best thing about it is that it's not a regular tutorial. It tells you what to build for each step, but it forces you to figure out how to actually do that on your own, which is the exact skill you need for a real job. With this project, you'll get hands-on practice with stuff like infrastructure as code, CI/CD pipelines and serverless technologies. But okay, you know your basics. You've done an introductory project like the cloud resume challenge. What should you do next? Well, step three is to keep building. Here, you need to keep the momentum going to build out a portfolio that is really hard to ignore. There's a few options you can take here. The first is to dive into AWS workshops. This is a massive collection of official free workshops designed by AWS to mimic real world problems. A particularly valuable section within this is the well-architected lab. These are specifically focused on teaching you best practices for building solutions. Learning to build things the right way from the start is a skill that will definitely impress interviews. But although the workshops themselves are free, you pay for the AWS resources that you use. Depending on the lab, some of this usage may fall within the AWS free tier. So you do have to be a bit more careful about cleaning up and deleting everything afterwards to control your costs. Learning to manage resources and costs in your own account is a valuable skill in itself. The second option is for those of you who already have a job or some experience with coding. This option is about finding something new to build, but about taking something you've already created and just rebuilding it for cloud. And the reason this is so powerful is because it has a real business context already. This is something that you've built that solves a real problem that a real business has faced. And this is what hiring managers want to see more than anything, that you can apply a cloud technology to solve a real problem. So what does that look like? Suppose your team has a Python script that generates some kind of report every night using data from Microsoft Excel. Right now it runs on a server every night on a cron job. Your challenge is to try and cloudify it. Maybe you could reengineer it to run as a lambda function triggered on a schedule by Amazon EventBridge.

with the final report being saved automatically to an s3 bucket. The same idea applies to your current job. And this is where you can find a lot of ideas. Look around you. Are there any ongoing cloud migration projects in adjacent teams? Does your own team have workloads that could be moved to the cloud? See how those opportunities and ask to be part of them, even if it's just in a small way. This kind of learners you go approach is how you truly build skills. And after this, it's just about repeating the same thing. Look out for more project. Make sure there's some business context attached to it. And then learners you build. I know the projects here aren't super in-depth. You know, binocs solutions that are going to be deployed to a large enterprise. But it should give you a good starting point and keep you going. If there's enough interest, I'll do a video in the future about more intermediate to advanced projects that you could do. So now that you have a clear plan to start building high impact projects, you're off to a great start, but you're still missing the most important step. You need to properly document and showcase these projects on your resume. So click here to learn the exact framework to use to show off your projects to hiring managers.

## **Summary**

You find a tutorial, you watch it, you follow all the steps, and then you immediately start looking for the next one . But a few days later, you can't remember any of the skills you actually learned . In a real cloud job, no one gives you a step-by-step tutorial. Instead, you have to actively search your memory or research the answer. When you hit a roadblock, your forester stop and ask, okay, so how do I actually solve this? That struggle is exactly what forges the strong lasting neural pathways.