

# THE CLOUD DEVELOPER

UNLEASH YOUR **CLOUD CAREER**

Moneer Rifai

# Table Of Contents

About Me	2
About You	3
Introduction	4
Overview of Required Skills	7
Skill 1: Cloud	9
Skill 2: Code	12
Skill 3: Tooling	15
Teach Yourself Like a Pro	18
Conclusion	23

# About Me

Hello! And thank you so much for downloading my ebook!

My name is Moneer, and I am a cloud developer and architect. I am assuming you are interested in the cloud and wondering how to break into this exciting and highly sought-after career. The good news: you came to the right place!

In 2017, after a series of dead-end jobs in IT, I decided to reinvent my career and break into the cloud. The process was not easy, since I had very little relevant experience. After much trial and error, and after almost failing, I was able to push through and find the winning formula that has led to the most fun, satisfying and exciting phase of my professional life so far.



My goal is to provide you with a formula for success that you can follow and in the process, avoid the many mistakes that I made. If you are committed, willing to teach yourself the necessary skills and not afraid of hard work, then you have a very promising and fulfilling career ahead of you!



# About You

Some of the assumptions that I will make about you:

- You are interested in the cloud and understand that it is the way of the future
- You are hoping to get a job working in the cloud and know that this expertise is in extreme high demand now
- You might have some experience in general IT or software development, but if you do not, you have the passion and the drive to learn
- You are interested in self-learning and willing to put in the effort
- You do not have the luxury of putting your life on hold to go back to school for a computer science degree or spend tens of thousands of dollars on an intensive bootcamp program

If some of the above applies to you, then this guide is for you.

Basically, this guide is for anyone who is interested in self-learning and wants to break into the cloud career, no matter your gender, age, background, technical expertise, or education level. If you are willing to put in the work, then this is for you.



# Introduction

The advent of the cloud has completely upended the way we live, work and do business. It has enabled accelerated innovation in ways never seen before seen. Many of the ubiquitous products and apps that we depend on today would not be possible without the cloud services that power them.

But what exactly do we mean by “the cloud?”

“The cloud” can refer to different things, but in our context, it refers to public cloud services that are open to anyone. The big three public cloud providers are AWS, Microsoft Azure and GCP (Google Cloud Services), but there are many other smaller players like Digital Ocean, IBM, Alibaba, Oracle Cloud, Linode, etc.

Fortunately for us, as cloud offerings got more popular and more complex, demand for cloud experts spiked. The goal of this guide is to help you become a cloud expert.

So how does one become a cloud expert? Let me illustrate by telling you about my personal journey.

# My Story

When I decided to switch careers back in 2017, I naively thought that all I needed was to pick a cloud provider, study their offerings, maybe pass one of their certifications, and I would be good to go.

That was exactly what I did: after a couple of months of studying, I passed all three AWS associate certifications, concluded that I was now a cloud expert, and started applying to jobs.

I got a few interviews, but to my surprise, they did not go well at all. I am generally a confident interviewer, but I was often completely failing the technical part. My AWS knowledge was useful, but it was only a small part of what employers were looking for.

What was my weakness? There were several, but my main weakness was a lack of coding skills.

I knew some Python and JavaScript but at a very basic level. Almost all my interviews involved some sort of a code challenge, and I did very poorly in that area.

After months of trying, a growing SaaS company in Chicago extended me an offer, despite my poor performance on their code challenge. I figured I had made it! I was now working as an SRE (site reliability engineer) on the development team!

My excitement quickly dissipated when I started on the job and found myself unable to keep up with even the basics. I was constantly overwhelmed and struggled with the complexity of the infrastructure and the code base. Months into the job, I still needed a lot of hand-holding.

Tension started to mount and the senior engineers were getting frustrated with me. It got to the point where I was confident I was going to be fired, so I took an unusual step: I told my team lead that things are not working out and that I should probably leave the team. As expected, she did not object. Three months into my first cloud job, I quit.

I was devastated. I felt that I had failed.

After I had some time to reflect, I realized where I went wrong. I had focused all my efforts on learning about the cloud and getting certified, but that did not fully prepare me for what jobs in the real world were looking for.

In this ebook, I will make sure that you avoid my mistakes and learn the skills that you need to be successful.



## Overview of Required Skills

As I mentioned earlier, simply learning about the cloud is not enough. That is the most common mistake that beginners make, and one that I made myself.

Besides learning about the cloud, becoming a competent developer is one of the most important skills that you will need to hone. I cannot emphasize enough how important this skill is.

In addition to learning cloud and code, there is a third category that I will refer to as “tooling.” This encompasses the endless list of popular tools and services that are often referenced in job postings. Some examples: Docker, Kubernetes, Git, Terraform, CI/CD, etc.

Many newbies get overwhelmed by the huge number of technologies that they feel they should learn. A common question I get is “I heard X is popular, should I learn it?” (where X is the name of a tool or a technology). My answer is: focus on learning cloud and code, and the tooling comes later.

For now, I want you to get a high-level view of what you need to learn. I think of these skills as the “three pillars” of modern cloud development.

## The Three Pillars of Modern Cloud Development

At a high level, you will need to master the following:

1. Cloud: you need to pick a cloud provider and become knowledgeable in their offering
2. Code: you need to pick a programming language and become a decent developer
3. Tooling: depending on your circumstances, pick a couple of tools to get familiar with

I will briefly discuss each one in the following sections.



## Skill 1: Cloud

I want this ebook to appeal to a wide audience, but I will make certain suggestions when it comes to choosing which technologies to focus on. With that being said, if you already have a preference for a specific technology (because your employer uses it or because you have experience with it, etc) then you should go with the choices that make the most practical sense to you.

When it comes to choosing a cloud provider to focus on, my suggestion (and what really worked well for me) is **AWS**. They are the market leaders by far and most importantly, AWS skills are in huge demand.

The rest of this section will focus on AWS but almost everything will apply to other providers like Azure or GCP as well.

## Cloud Certifications

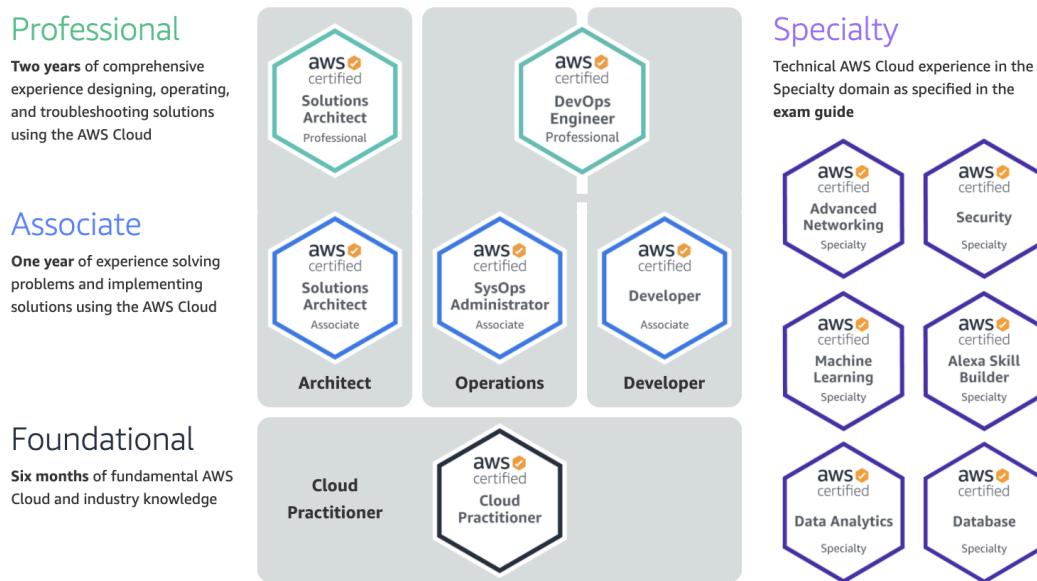
Certifications are a controversial topic in technology, but in this ebook, I will encourage you to pursue them.

The value in achieving a certification is not earning the title itself but rather the fact that a certification program gives you a structured and easy path for learning that you can follow.

I cannot underscore this enough: many beginners think that passing a test and getting certified will make them qualified for being a cloud developer or DevOps engineer, but the reality cannot be more different.

Certifications alone are not enough, but they are a good place to start.

If you have some background in IT, you might find yourself able to easily pick up on some of the cloud services that are covered in certification exams. For most people, pursuing one or more of the associate level certifications is a great place to start. If you have no background at all, AWS has an entry-level certification that would be a good option. See AWS's certification path below for more information.

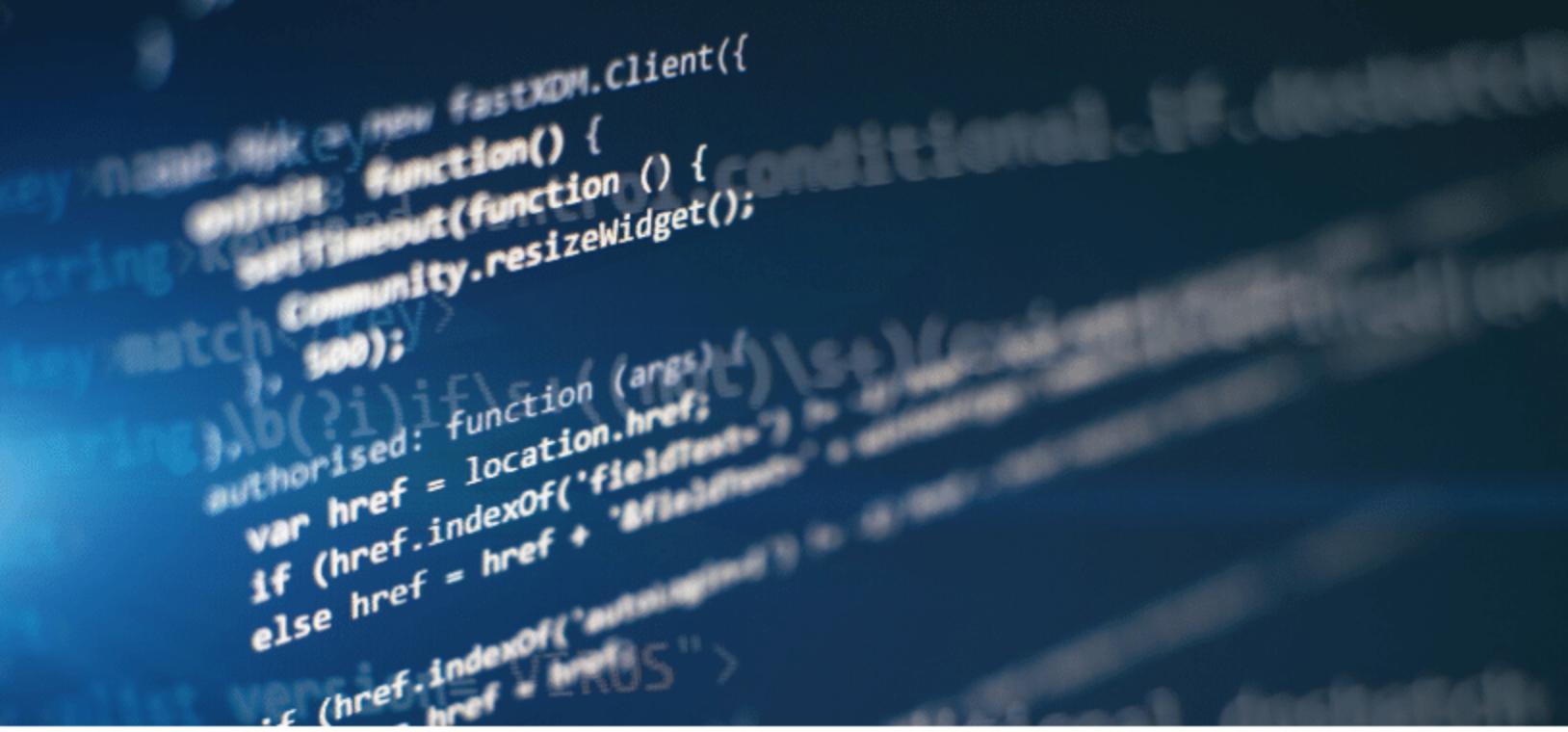


## Certifications Alone Are Not Enough

I already said this, but it bears repeating: simply passing a test is not enough. Be careful of leaning too heavily on your certification because, to be honest, most hiring managers do not care whether you are certified or not.

But if you have a couple of AWS certifications and supplement that with additional skills, then you become a very strong candidate who is qualified for most cloud-based jobs.

In the next section we will discuss the second pillar: code.



## Skill 2: Code

There are many people out there who have a background in IT and can easily pick up some cloud skills, but it is less likely to find someone who has cloud skills and who can also write code. This is the winning combination that will unlock endless opportunities for you.

Additionally, if you are not comfortable writing code and thinking as a developer, you will find most of the tooling very difficult to grasp, because most of these tools assume a certain comfort and expertise with programming.

When it comes to choosing a language, just like choosing a cloud provider, if you have a reason to focus on a specific language (maybe your employer already uses it and it would be advantageous for you to learn) then do it. If you have no preference, then my suggestion is **Python**.

Python is one of the best languages for the cloud, it has a huge community and it is fairly easy to learn. But do not let this fool you: Python is extremely powerful and has endless applications, so becoming good at Python will be a huge boost to your career.

When it comes to learning programming, there are no certification programs (not reputable ones anyway), so learning a programming language is just a matter of doing a lot of practice and a lot of writing code. You will need to get your hands dirty; it is impossible to learn if you do not practice.

## How Learning to Code Saved My Career

As mentioned earlier, back in 2017 when I wanted to get into the cloud, I focused on getting my cloud certifications and studying AWS and neglected to improve my coding skills. As a result, I failed at most job interviews and the one job that gave me a chance, I was not able to keep up with.

The primary reason was that I simply did not know how to write code, and how to think as a developer.

After realizing where my weakness was, I proceeded to fix it. I picked Python and tried to focus my full efforts on learning the language and getting experience with it.

The process was not easy. At times it was tedious and frustrating, but after finding the right learning resources (more on that later), I started to feel that I was making progress. Every day I would learn and work through quizzes, tutorials and exercises. Most importantly, I was actually writing code (and not just learning about it), and that hands-on process started to reinforce my knowledge.

Then one day, out of the blue, a recruiter reached out to me on LinkedIn. He was looking for someone with AWS experience who could also write code in Python. We set up an interview.

The technical interview started out well and then we moved into the code questions. My heart was about to leap out of my chest, because that's where I usually crash and burn. Not this time! I was able to answer most of the code-related questions, and I felt a huge sigh of relief. But just when I thought that I aced the interview, my interviewer proceeded to tell me that the next step will be a take-home coding challenge that I will have a week to complete.

The coding challenge was fairly complex, and it involved technologies that I have never seen before (like AWS Lambda and CloudFormation), but I found myself able to pick it up fairly easily! My coding knowledge enabled me to quickly learn new technologies enough to be able to use them. Needless to say, I completed the challenge, was offered the job, and three years later, I am very much still happily employed there!

In a later section, I will share learning resources for Python but for now make the commitment to become a strong developer, without which it will be very difficult to succeed.



## Skill 3: Tooling

There is an endless number of tools that one could learn, and it seems that new ones keep popping up. So what do you prioritize?



*There is an overwhelming amount of cloud tools*

First, it is important to differentiate between the tool and the underlying technology.

For example, Github is a popular version control tool, but the underlying protocol is Git. If you know Git, you will be able to quickly pick up Github, or Gitlab, or other Git-based tools.

Second, you do not need to learn every tool out there. Even if a job description references a certain technology that you are not familiar with, that does not mean you are not qualified. Most job descriptions are wish lists, and it is almost impossible to find someone who has experience in every tool and technology that some employers advertise as “required skills.”

## Think Categories, Not Individual Tools

There are countless tools and products out there, and I am certain that more have come out since I wrote this guide. Generally speaking though, they tend to fall into a few categories. The following is a non-exhaustive list:

- Collaboration: tools like Slack, Jira, Trello, etc.
- Code Hosting and Versioning: this includes Github, Gitlab, Bitbucket, etc.
- CI/CD: includes products like CircleCI, Travis CI, Jenkins, Gitlab, AWS Code Pipeline, and many others.
- Configuration Management: some examples are Puppet, Chef, SaltStack, Ansible.
- Monitoring, Logging and Alerting: tools like Nagios, LogicMonitor, DataDog, Prometheus, CloudWatch, PagerDuty, NewRelic, and many others.
- Containerization: this category might be unfamiliar to those who come from a traditional IT background, but with the popularity of Docker containers, a new tech category has emerged. The two most popular products in this category are Docker (for containers) and Kubernetes (for container orchestration).
- Infrastructure as Code (IaC): popular frameworks are CloudFormation (for AWS) and Terraform (cloud-agnostic).
- Operating Systems: many jobs will expect you to manage virtual servers and other systems. Linux is the most popular, but many jobs also use Windows Server. Using Bash or PowerShell, respectively, is the most common way of interacting with these servers.

There are more categories. But these are the main ones.

## Which Tooling Categories Should You Learn

It depends on how much time you have, but my advice is to focus on the first two pillars (cloud and code) and worry about Tooling less.

You see, I believe if you are good at writing code and you understand cloud infrastructure, many of these tools will come easily to you. On the other hand, most of them will be difficult to wrap your head around if you do not have a strong command of cloud and code.

For example, if you know Python really well (or a similar language), you can pick up something like CloudFormation or Terraform in a single afternoon. The same holds true for most of the other tools mentioned above.

Consider the following as general guidelines:

1. If you are pressed for time or if you come from a non-technical background: just focus on getting really good with a cloud provider and becoming a competent developer in one language
2. If you have some time and feel good about your progress on the first two pillars: consider learning Git and an Infrastructure as Code framework. Also consider learning about containers because they are very popular. Brush up on your OS of choice.
3. If you want to take it to the next level: learn CI/CD

Whew! I hope I did not overwhelm you with all the things that you have to learn. At the risk of sounding like a broken record: focus on learning a cloud provider and a programming language first, and everything in this chapter comes second.

But how and where do you learn all this stuff? Well, that is the topic of the next chapter.



## Teach Yourself Like a Pro

This chapter is what I consider the “meat” of the program.

I will share with you some of the best resources for learning as well as a possible program that you can follow.

### Learning Resources

There are many great learning resources out there, like books, blogs, videos, courses, etc.

Because of how fast this area of tech changes, it is important that you are relying on the most up-to-date resources.

I have experimented with almost every learning resource out there, and I have found that video courses are by far the best approach. More specifically, courses on the Udemy platform will be your best bet. Note: I am not affiliated with Udemy in any way.

Udemy always has a sale so you can get any course for around \$10-\$12. They have many high-quality courses with content that would otherwise take you months to cover. I learned Python and passed all three AWS certifications from Udemy courses.

## AWS Certification Courses on Udemy

My recommendation for studying AWS is:

- courses by A Cloud Guru (they also have their own platform, but they still publish courses on Udemy)
- courses by Stephane Maarek, who is the most popular AWS instructor on Udemy

## Python Courses on Udemy

For programming content, there are lots of options out there. Some of my favorite instructors are:

- Colt Steele (his Python course is highly recommended)
- Andrei Neagoie
- Jose Portilla
- Tim Buchalka (his Python course is massive)
- Al Sweigart (*Automate the Boring Stuff with Python* is wonderful and entertaining)

## Tooling Courses

For other topics, like the various areas of Tooling, Udemy has a wealth of courses on any given topic. You will definitely find high quality content for your topic of choice.

## Learning Path

The following is a sample path that you can follow, but feel free to adjust according to your previous knowledge and experience (or lack thereof).

Your Path:

Learn AWS and Python, then pick up a cloud tool or two if you can.

## Your Goals:

- Pass the AWS Certified Solutions Architect – Associate Certification
- Pass one or both of the two remaining associate-level AWS certifications
- Learn Python
- Learn a cloud tool (recommended: Git)
- Learn another cloud tool (example: Docker)

## Your Approach:

Start by simultaneously studying for the AWS Solutions Architect certification (associate level) and learning Python. Continue to learn and practice Python as you pass one or more certifications, and as you move into learning about cloud tooling.

## How Long Will This Take You

How much time you need will vary greatly on how much experience you already have and how much time you are willing to invest.

When it comes to learning, I highly recommend you establish some sort of a daily practice, and if you have the time, you can do longer study sessions over the weekend. Practicing daily will help reinforce your learning, so for example learning for one hour per day is much more effective than learning for seven hours at a time on Sundays.

With enough focus and dedication, you can potentially finish the above path in anywhere from a couple of months to a year. What took the most time for me was jumping around different areas of learning and not having a plan. Having a plan will definitely streamline the process.

## Plan Your Journey

I cannot emphasize how important it is to have a plan. Block off a daily chunk of time that you will dedicate to learning. Do not assume that “you will find the time” because you won’t unless you block that off.

If you are a parent or are generally very busy, you will definitely have to make some sacrifices. This might mean socializing less with your friends or cutting out certain activities like TV and entertainment. I truly believe that no matter how busy you are, if you sit down and audit where your time goes, you will find opportunities to reclaim time for studying and learning.

## Learn How to Focus

Knowing how to be laser-focused on the task at hand will be one of the most important skills that you will need to develop for learning now and later on for a successful career.

Cognitively demanding subjects like coding and cloud infrastructure require focused attention. Learn to improve your focus by removing distractions and turning off phone and social media notifications.

Practice the Pomodoro technique. It changed my life.

Read *Deep Work* by Cal Newport. It also changed my life.

## Learn How to Persevere

I will be honest with you: this process could be difficult. It looks glamorous, but day-to-day you will be faced with frustration, boredom, error messages and complicated abstract ideas. Knowing how to persist through this and push through is a critical skill that you will need to hone.

Sticking with the plan even when things get tough is easier said than done. This also means avoiding the temptation to pursue something else because what you are learning is becoming boring and difficult. This is something I struggled with. Like many others, I can easily be a victim of my own shiny object syndrome. I start down a path with a lot of excitement, but when it gets difficult, I get bored, and then I start pursuing something else.

I spent the bulk of my time off back in 2017 doing exactly that. The good news: it gets easier with time. Persevering at first is hard, and you have to force yourself to do it. Resist the urge to explore other things when you get bored. Be comfortable with boredom. You will eventually get past the urge to chase something else more shiny when you run into issues, and you will develop a sense of pride and accomplishment in the process.

Keep your eyes on the prize, push through the obstacles, and you will be amazed at how far you'll go.



## Conclusion

If you have read this far, then you are well on your way to unleashing your career, and I might add, transforming your life.

I will not promise you that this journey will be easy. It requires perseverance, dedication and commitment. What I can promise you though is that you will not regret having embarked on it.

Please feel free to reach out. If you downloaded this ebook from my website then you already know how to connect with me. I look forward to hearing from you, helping you when you get stuck, and congratulating you when you achieve your goals.

But now it is time for you to start writing code and building some cloud infrastructure! Best of luck.