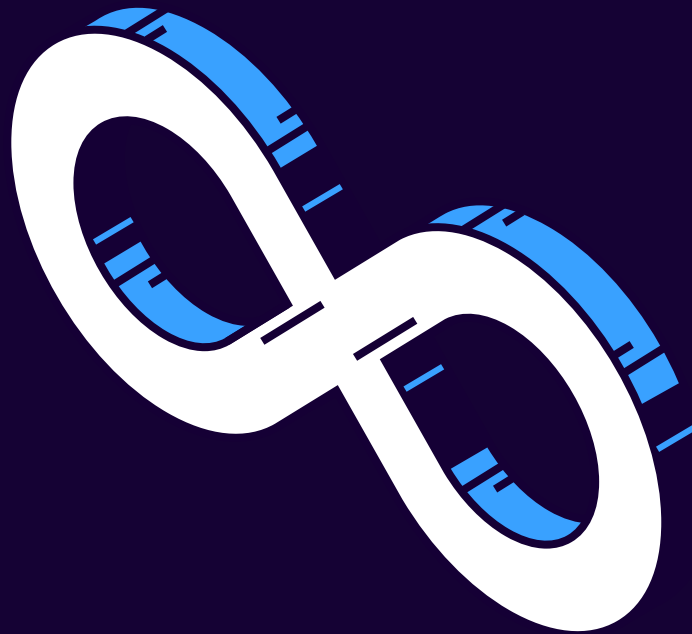


# DevOps

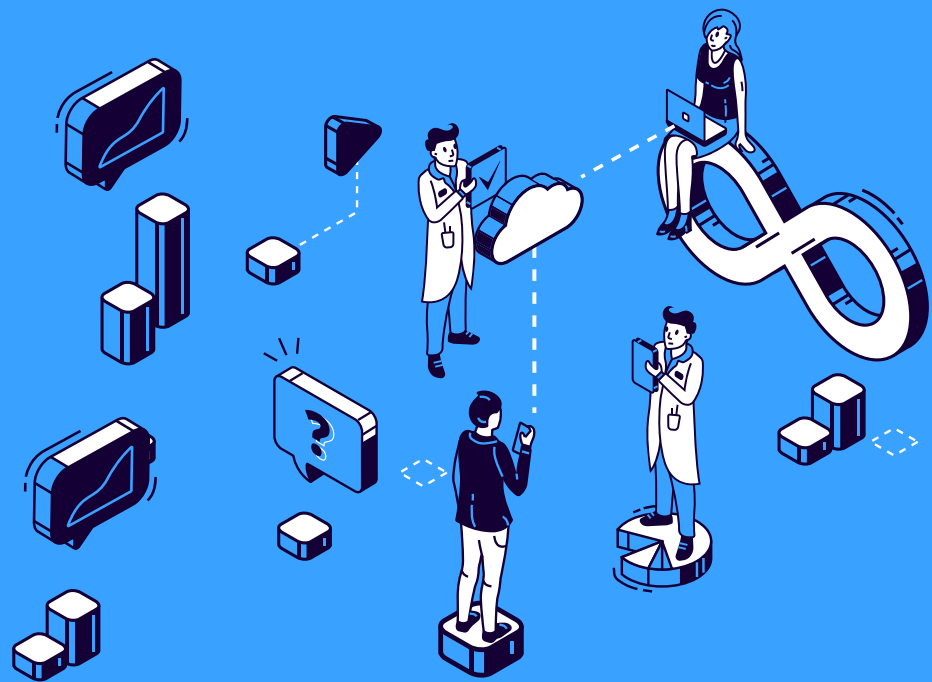
From Concepts to Practical  
Applications

---



# Table of Contents

Overview	01
Choosing the Right Deployment Model	02
Applying a mature DevOps approach to your Software Delivery	06
The Tools for the Job	07
Scripts and Automation	08
Cloud Services Going “All In” for DevOps	11
How Microsoft Azure is making it easier for DevOps teams to work with each other	12
How AWS is approaching DevOps	15
DevOps from other Cloud providers	16
The Future of DevOps	17



Software delivery is more relevant in a digital-first world than it has ever been. Customers require solutions that are developed quickly, evolve to changing demands, run reliably, and, above all, are secure. This set of requirements firmly supports the modern DevOps delivery model. In this ebook, you will explore the following:

- ✓ The evolution of different delivery models
- ✓ The core tools required to build out a foundation DevOps team
- ✓ The need to rely on automation and scripting
- ✓ How Cloud Service providers are going “all in” to support DevOps delivery
- ✓ A closer review of the specific DevOps tools that AWS, Google Cloud, and Microsoft Azure are delivering to support DevOps
- ✓ The future of DevOps

When you reach the end of this ebook, you will have the knowledge needed to build a DevOps team and start working on your first projects.



# Choosing the Right Deployment Model

Software delivery is complex. There are many tools on the market that make it easier to deliver solutions. Each approach to managing the work has its strengths and weaknesses.

The bottom line is that as soon as the software implementation becomes easier, then the demands of your customers will increase. Over the last two decades, three models have emerged as the default paths for software delivery: Waterfall, Agile, and DevOps.

- ✓ **Waterfall:** this is the oldest model supported by Kizen and PMBOK (Project Management Book of Knowledge) among other solution delivery models that date back to the 1950s
- ✓ **Agile:** the rise of the Web and the demand for faster releases of software drove the evolution of Agile, a model where you can deliver solutions quickly to your audience
- ✓ **DevOps:** the modern digital age, powered by Apps and services, must scale to support billions of uses. These solutions need continuous delivery of new features and support resulting in what is now known as DevOps

Each of these delivery approaches come with their benefits and detriments.

## When To use Waterfall

The oldest model for software delivery is Waterfall. The term “waterfall” comes from the cascading approach applied to delivery. For instance, workflow through the following stages:

- 1. Business Need**
- 2. Business Process Assessment**
- 3. Development**
- 4. Testing**
- 5. User Acceptance Testing**
- 6. Quality Control**
- 7. Release**

The start and end of each stage are usually accompanied by a set of Gateway Approvals to ensure that the needs of the next stage have been met. The process is long and clearly breaks apart the different groups working on the project.

It is easy to bash Waterfall. The approach is process heavy and frequently ends in failure (depending on how you determine failure). However, many of the tools used in Waterfall have value. The Kaizen (Japanese for “improvement”) mindset drove the rapid growth of companies such as Toyota. To this end, there are times when Waterfall is of value. Older systems (often referred to as Legacy) usually are good candidates for Waterfall. The solutions are made of large blocks that make it difficult to split.

## Leveraging Agile for modern delivery

In many ways, Agile and Waterfall are strongly related to each other. The main difference is that in Agile, cross-functional teams are pulled together and are asked to iterate quickly on solution delivery. Unlike Waterfall, Agile will pick a team together comprised of the following skillsets:

- ✔ Business Analyst
- ✔ Development
- ✔ Security
- ✔ QA
- ✔ Release Management

If you have a solution that is part of the post-Web era (i.e., was built any time in the 21st century), then the architecture of the solution will accommodate Agile delivery. Agile makes it easier for a team to “sprint” and deliver working code over a two-week cycle.

As with Waterfall, there is an increasing number of models and resources you can leverage to activate your Agile team. Models such as Scrum are well documented and give any leader the blueprint knowledge on how to drive an Agile squad forward. Additionally, tools such as Jira and Trello makes it easier for teams to manage their work.

Agile delivery is an excellent place to start if your company has only ever done Waterfall projects and want to explore approaches that deliver faster results.



# Applying a mature DevOps approach to your Software Delivery

There are three spaces to follow when it comes to maturity for DevOps:

- ✔ Open Source Tools will continue to dominate
- ✔ Skills to deliver DevOps solutions will be more freely available
- ✔ Delivery of DevOps will become commonplace

Open Source Tools are the heart and soul of DevOps. Whether you are using Jenkins, Kubernetes, Chef, or programming with PHP, the goal for many years has been the adoption of Open Source solutions. There is one overriding reason why Open Source is so essential for the future maturity of DevOps: trust. As a software engineer, you have many choices. The one that you will gravitate towards is the one you trust. Open Source lays bare how the solution is built. A great example of an Open Source solution that is now mature and has a trusted audience is Kubernetes. The Open Source program continues to gain momentum and provide the tools needed for scalable networks running in the Cloud.

A second key area to watch in 2020 is the maturing of programs, such as those run by Simplilearn, to offer a broad range of training to upskill people who want to be part of DevOps. Practice for GitHub, Docker, and Ansible can now be found collected together in structured programs. You no longer have to hunt the internet to find articles and videos that loosely link DevOps skills together. Now you can feel confident in the courses you take to upskill your resume and be a valued member for any DevOps team.

Many companies are still struggling with rapid delivery. A stop-gap method for many is the adoption of Agile delivery. Agile is a good step away from Waterfall delivery, but common problems such as testing and consistent delivery still plague Agile. Companies that had been experimenting with DevOps in 2018 and 2019 are now moving to have DevOps be a delivery model supported by the CIO. The cost of running a DevOps team is going down as the tools skills and tools become more readily available. In 2020, you will start seeing DevOps teams driving mission-critical functions for companies.



# The Tools for the Job

The DevOps cycle has essentially two sides to it: the Development side and the Operations side. For both sides, there are specialized tools. What makes DevOps different from previous deployment models is that both sides are integrated. Each tool the Developer uses is designed to support the tools of the Operations team and vice-versa. Therefore you will find the commonalities such as security, automation and modular design are prevalent with all of the DevOps tools. Finally, almost all DevOps tools are Open Source. Indeed, some of the best tools for your DevOps teams are developed and shared by companies such as Google, Microsoft and AWS.

Below is a list of some of the tools you will need to get your DevOps team started. To be clear, this is not an exhaustive list. There are hundreds of tools your DevOps teams can leverage.

- ✔ **Docker:** a virtual environment where you can build and script solutions that can be shared easily with your development team for continuous deployment
- ✔ **Git:** advanced source code management
- ✔ **Maven:** continuous build tests that can be used to validate code and functionality in a solution
- ✔ **Jenkins:** continuous integration from developer to production
- ✔ **Ansible:** continuous deployment to production environments
- ✔ **Kubernetes:** continuous management of containers in a production environment

The keyword “continuous” is common with the tools you will use as a DevOps leader. The focus for any DevOps team is to build small blocks of code that can be effectively tested and then deployed to production environments.



# Scripts and Automation

Arguably the most compelling reason why you will want to take advantage of DevOps is that the mindset, tools and delivery require your team to think about scale. Scaling is accomplished with two key factors: Scripts and Automation.

Each of the tools listed above can be scripted. You will need to work with each tool to understand their scripting language, but most fall into one of three languages:

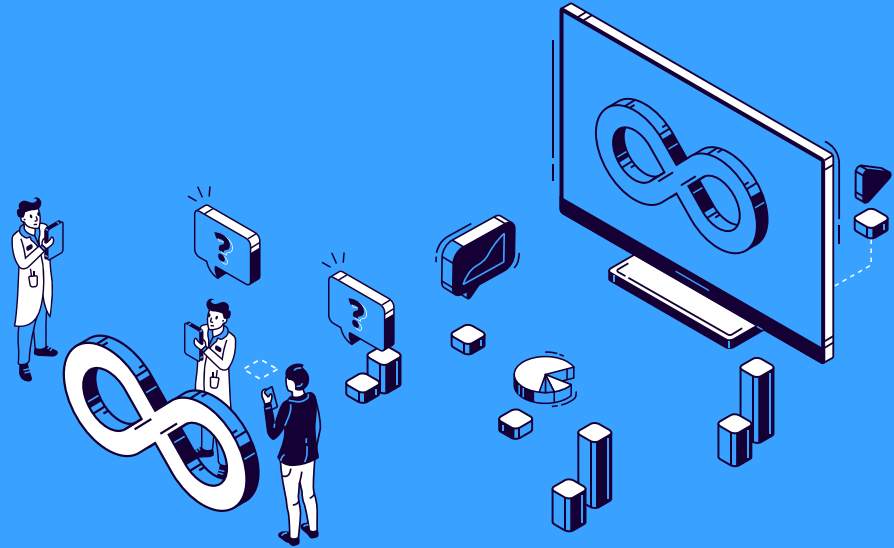
- ✓ JSON
- ✓ YAML
- ✓ XML

All three are simple scripting languages that can be used to describe what the tools need to accomplish. For instance, the following is a sample YAML script for Ansible:

```
---  
- hosts: webservers  
  
vars:  
    http_port: 80  
    max_clients: 200  
    remote_user: root  
  
tasks:  
    - name: ensure apache is at the latest version  
      yum:  
        name: httpd  
        state: latest  
    - name: write the apache config file  
      template:  
        src: /srv/httpd.j2  
        dest: /etc/httpd.conf  
      notify:  
        - restart apache  
    - name: ensure apache is running  
      service:  
        name: httpd  
        state: started  
  
handlers:  
    - name: restart apache  
      service:  
        name: httpd  
        state: restarted
```

Once you have a script written, you can now run the script whenever you need to. For instance, you can have Kubernetes rebuild a service if it fails. In addition to automatically running scripts, you can also trigger scripts from command line tools. Also, you can use the chat tools in Slack and Teams to run scripts for your DevOps environment. For instance, each time that Jenkins runs, it can send a notification to a group chat window in Slack so that the whole team is fully aware of the completed process.





# Cloud Services Going “All In” for DevOps

In many ways, the Cloud is a perfect environment for DevOps. Each service is modular, scalable and securely connected. For this reason, leading Cloud providers are using their services to attract DevOps teams and to encourage companies to stand up DevOps teams. The leading Cloud providers, Microsoft, Amazon and Google, all offer similar products that include:

- ✓ Data storage
- ✓ AI
- ✓ Databases
- ✓ Web tools

In addition, you will also find a Marketplace with each vendor where you can plugin third party products such as Kubernetes, Jenkins and Docker. In many ways, these offerings are table stakes, the bare minimum you will need to use DevOps in the Cloud. You should expect a top tier vendor to provide these and other DevOps services. What separates the leaders from each other is how they are doubling down on specific tools for DevOps teams. One company, Microsoft, is leading with new DevOps tools they are releasing.

# How Microsoft Azure is making it easier for DevOps teams to work with each other

Microsoft has many tools for developers. A good place to start is <https://azure.microsoft.com/en-us/services/devops/> where a list of the Azure specific DevOps tools can be found. The current list includes:

- ✓ Azure Boards
- ✓ Azure Pipeline
- ✓ Azure Repos
- ✓ Azure Test Plans
- ✓ Azure Artifacts

Azure Pipelines is one product that is particularly interesting. Microsoft introduced Azure Pipelines with a single vision: Make connecting the tools you need in DevOps for Continuous Integration and Continuous Delivery (CI/CD) easier to use. This approach translates to connecting Docker with Git, Selenium, Jenkins, Chef, and the many tools you are currently using in your DevOps stack. Azure Pipelines are the glue that holds everything together, which is quite a feat when you think about it.

As you might expect, Azure Pipelines is optimized to run solutions built for Microsoft's Azure Cloud Service (the hint is in the name). However, it's very refreshing that Microsoft is not restricting Azure Pipelines to only Azure solutions. Traditional enterprise solutions running behind a firewall can also take advantage of the Azure Pipelines toolset. In an interesting move, [Microsoft enabled Azure Pipelines to connect from one cloud provider to another](#), making it act more like an open-source tool than a proprietary one. This capability lets you use configurations that include a hybrid cloud, a private cloud, or even a competing provider including Redhat, AWS or Google Cloud.

The goal with Azure Pipelines is to make DevOps easier to set up for teams. If your team is new to DevOps, you can now get running and testing DevOps delivery models in hours versus days; we all know that with emerging technology, speed makes an impact.

## What Azure Pipelines Gives You

As you might expect, Microsoft is making Azure Pipelines user-friendly. To begin using Azure Pipelines, a good place to start is to take an Open Source project you are hosting in GitHub and connect it to Azure Pipelines. Here are the steps you should take:

8. Register at <https://dev.azure.com> to obtain access to a platform that will allow you to manage your apps. The project can be public or private (with you controlling access).
9. After you've completed your registration, connect Azure Pipelines with your project. You can connect to any Azure Repos via Git, BitBucket, GitHub, GitHub Enterprise, any 3rd party Git server (such as GitLab), and Subversion.
10. Once you've connected Azure Pipelines with your project, it's time to configure your Azure Pipeline. There are two options; create a default starter pipeline or use an existing Azure pipeline. Using an existing Azure Pipeline should be used mostly by experienced teams only.
11. Once complete, review the script that will manage your Azure Pipeline activity.

And that's it! Now you can run your Azure Pipeline and, as you would expect, you

will be given outputs that validate that your code has been tested and built. In less than an hour, you can have your first Azure Pipelines environment up and running.

## The Difference Between VSTFS and Azure Pipelines

If you have been using Visual Studio Team Foundation Services (VSTFS) then you might be thinking: "Hey, Microsoft, did you just rebrand VSTFS to Azure Pipelines?" And to a certain degree, you would be right. Azure Pipelines is based heavily on VSTFS simply because VSTFS is a strong and mature suite of products.

The big difference comes to light when you look as to how the tools in Azure Pipelines work together to deliver a DevOps model for delivery. In the last couple of years, VSTFS has grown to be two different products:

- ✔ An on-premise/enterprise version (legacy VSTFS tool suite)
- ✔ A cloud version

Azure Pipelines has more in common with the cloud version of VSTFS.

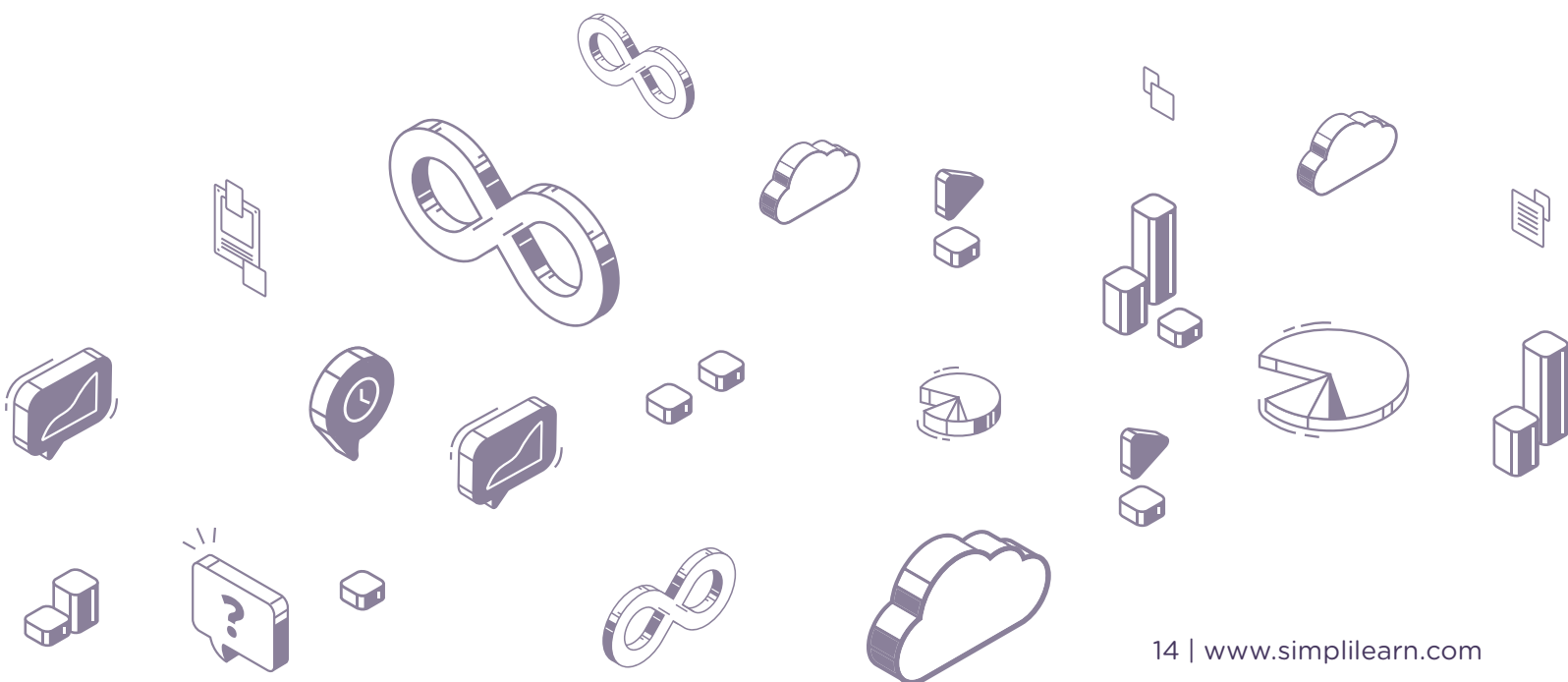
With all of this said, Azure Pipelines is maturing very quickly, providing a comprehensive and efficient suite of tools that can be easily extended with 3rd party plugins needed to meet today's DevOps delivery demands.

## More DevOps is Coming to Azure

A great deal of news is emerging from Microsoft on DevOps. To keep up to date on the latest announcements and trends, I recommend that you check the following sites for regular updates:

- ✔ **The Official DevOps Blog** - Here is everything you could possibly want to know about Azure and DevOps in one place. This site displays the related news and updates on the current sprints underway with the Azure team.
- ✔ **Azure DevOps YouTube Channel** - This stellar YouTube channel features everything you could want to know about DevOps products by Microsoft. Learn about the products and interact with a vibrant community that is integrated with Azure as well.
- ✔ **Azure DevOps Documentation** - Feeling like you need some in-depth attention to Microsoft DevOps? This technical documentation is all that you need to learn the nuances of Azure DevOps.

While these resources are a list of the main resources available for Azure DevOps, there are also many great Twitter and LinkedIn accounts which you can track down. As your skills expand, hopefully, your network of resources does also.





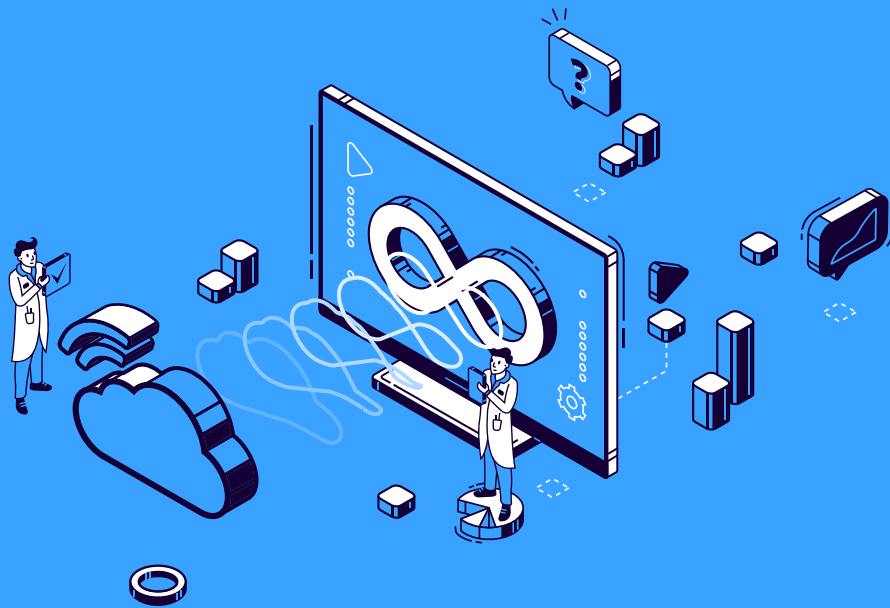


# How AWS is approaching DevOps

Amazon's AWS does support the full stack of DevOps tools you would expect to see. There are no big surprises. You can even take DevOps certification with AWS tools.

What is surprising is the direction that AWS appears to be taking as they deliver more tools for DevOps teams. The focus appears to be on low code or no-code solutions. A low code or no-code tool requires no scripting. You can be up and running with little or no knowledge of programming. This lack of coding capabilities is a blocker for DevOps teams who must be able to script their tools easily.

The good news is that AWS has continuously shown that they can respond quickly to market trends. If they see a need to build additional tools to support DevOp team creation, then expect that they will.

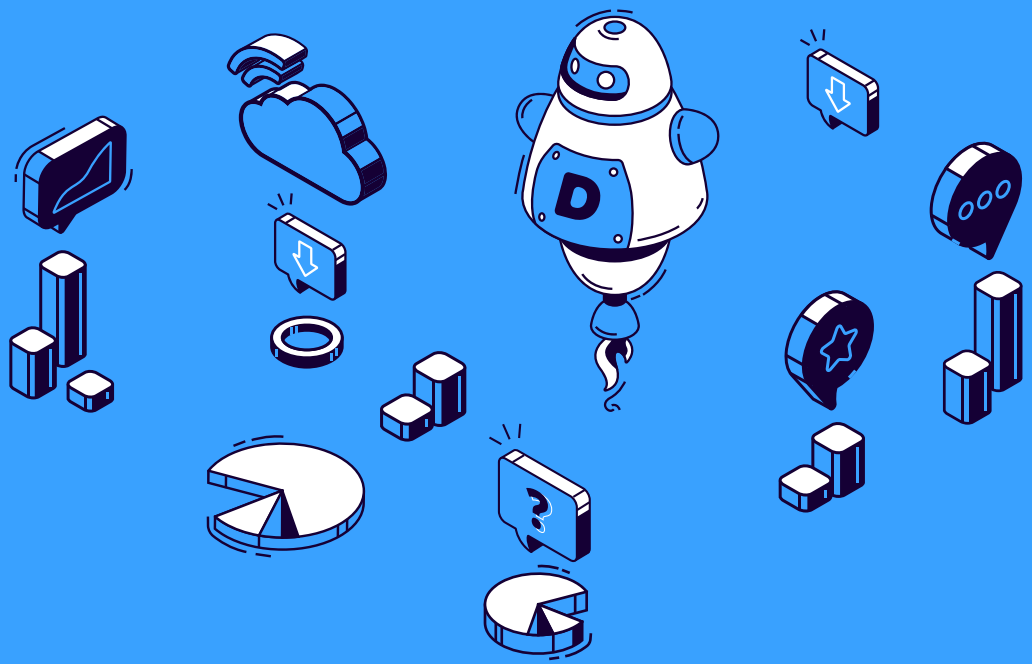


# DevOps from other Cloud providers

There are many cloud providers, and with the increased focus on Cloud vendors working more easily with each other, you can expect more specialization in tools. In addition to Azure and AWS, other vendors you will want to keep a close eye on include:

- ✓ **IBM/Redhat:** In 2019, IBM acquired Open Source giant Red Hat. Red Hat has a strong relationship with the DevOps community, and now IBM is looking to bring that same focus into the main company
- ✓ **Google Cloud:** Google is placing a focus on AI and Development with their cloud tools. DevOps is central to their culture and should be watched closely
- ✓ **GitLab:** A good alternative to GitHub for managing projects in the cloud

Expect the growth of DevOps in the enterprise to be reflected in the support cloud providers will offer.



# The Future of DevOps



#### **INDIA**

**Simplilearn Solutions Pvt Ltd.**

# 53/1 C, Manoj Arcade, 24th Main,  
Harlkunte  
2nd Sector, HSR Layout  
Bangalore - 560102

Call us at: 1800-212-7688

#### **USA**

**Simplilearn Americas, Inc.**

201 Spear Street, Suite 1100,  
San Francisco, CA 94105  
United States

Phone No: +1-844-532-7688

---

[www.simplilearn.com](http://www.simplilearn.com)