



DevOps platform suite comparison guide

Visit www.ten10.com

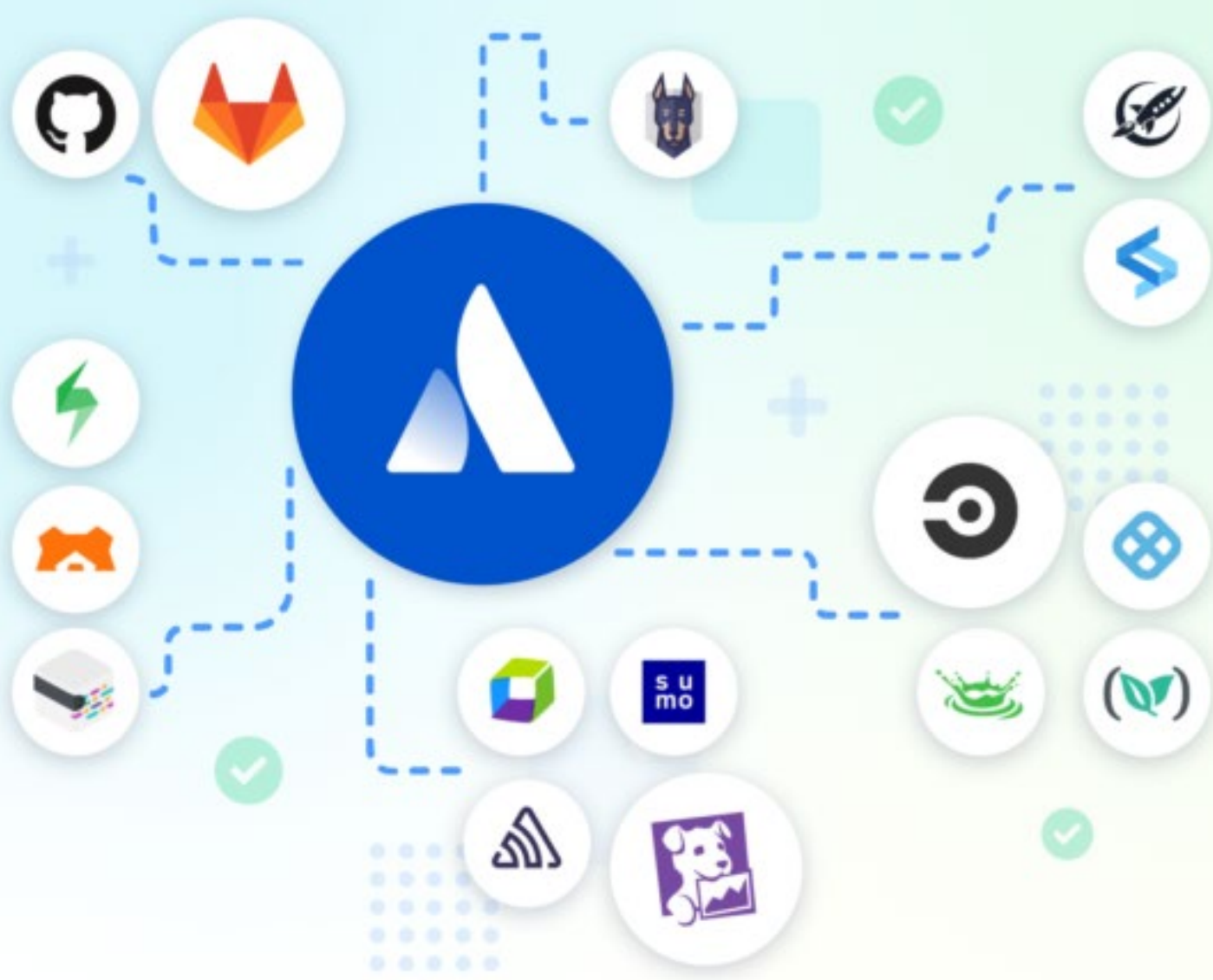
Ten10

Learn what comprehensive tool suites are available for your DevOps transformations

If you've spent any amount of time learning or doing DevOps, you'll know the vast number of tools and platforms at your disposal. It can be a minefield, with everyone vying for your attention. Most engineers will have a preference based on previous experience, but should that justify such a big decision? Here we aim to provide a comparison of some of the major platform tooling providers available to you.

For the purposes of this comparison, we are looking predominantly at providers of a comprehensive suite of tools. There are plenty of alternatives that focus specifically on areas such as monitoring, alerting, testing, documentation, logging, etc. Each of the providers we compare aims to facilitate a large proportion of the software delivery lifecycle, including (but not limited to) source control, task management, CI/CD, artefact storage and documentation.

Think of this guide as an overview of the major tool suite providers so you know exactly what they provide for different stages of the software development lifecycle. By knowing how these differ, you can make an informed decision on what to adopt before you start your DevOps journey.



Atlassian

Atlassian is the favoured service provider for many software businesses due to the excellent integrations between its core services. Atlassian supports Google Workspace integration, allowing for SSO across all products. This makes many of the integrations appear seamless. It has a wide range of products, many of which we won't address here. Its core offerings are as follows: cost overruns on your overall budget and resources. They can significantly affect your financial projections, lead to resource strain, and potentially disrupt other projects or initiatives due to resource reallocation. By analysing these impacts, you can develop a more strategic approach to managing your cloud costs, ensuring that unexpected expenses don't derail your IT operations or your company's financial health.

Source control – Bitbucket

Bitbucket is actually one of the older version control services, launched in 2008. There is a Cloud and a Server offering, so you can either self-host or let Atlassian manage it for you. It supports many of the frequently used Git features including pull requests, merge checks and code searching. They also support static file hosting on Bitbucket Cloud, providing a URL within the bitbucket.io domain.

At the time of writing, Bitbucket Cloud has a free tier, for up to five users. Beyond that, it's \$3 per user per month for Standard, and \$6 per user per for Professional (see tier comparison for more info). It's worth noting that Atlassian is in the process of discontinuing the server offering, with an expected end of support for server on February 15th, 2024. It is being replaced by Bitbucket Data Centre, a similar product but aimed more at enterprise users.

Our score: 4/5

Documentation – Confluence

Confluence is the Atlassian answer to collaborative knowledge sharing. It comes with a wide range of templates for different types of documents and allows for intelligent project hierarchy and organisation.

The concept of 'spaces' allows for different teams to work on projects independently of others, and the access control is very versatile allowing for granular control over who has access to what. As with most of the other Atlassian products, the search functionality within Confluence is very good.

Similar to Bitbucket, the pricing model starts with a free tier, limited to 10 users. There is also a Standard and Premium tier, running at \$5.75 and \$11 per user per month respectively. At scale, this can result in fairly heavy costs.

Our score: 4/5

Pipelines – Bitbucket Pipelines and Bamboo

Most of the pipeline offerings here operate in a similar way. You provide them with a config file in your source code, that defines when they should be triggered, and what they should do. Then, when that particular action occurs (usually a commit/merge to a particular branch, or the raising of a PR) the pipeline is triggered. It then runs a container, copies the source code into the container, and runs whatever you have told it to run against that version of the source code.

It's billed by the amount of minutes per month your pipelines take to run. For Bitbucket, the number of minutes you have available to you is dependent on the pricing tier. For the free tier, you get 50 minutes per month (probably not a lot if you have a very active codebase).

Since it's a Bitbucket feature, it's not possible to trigger the pipeline from anywhere other than a Bitbucket repository. If you're using some other source control provider then you likely won't be using Pipelines.

Bamboo is the Data Centre (on-prem) offering. Instead of containers, you install an agent somewhere. You are billed per agent. At the time of writing, 1 agent is \$1,200 annually so it's definitely aimed at enterprise users.

Our score: 3/5

Task management and bug tracking – Jira

For a long time, Jira has been considered the go-to project management tool for software engineering companies. It has been around since 2002 and supports different agile ways of working (most commonly scrum and kanban). It is an excellent choice for tracking large projects with visibility requirements spanning all areas of the business.

In recent years, several competitors have emerged, in part due to the complexity of Jira. The user interface can be a bit difficult to navigate at times and the ability to generate custom workflows, whilst seemingly versatile, can lead to maintenance overheads over time.

Our score: 4/5

Incident management – OpsGenie

OpsGenie was purchased by Atlassian in 2018, though as a product it is well-established and rich with features. It covers alerting and notifications, on-call scheduling, escalation policies and analytics/reporting. One benefit of being brought under the Atlassian umbrella is that it integrates with the other products seamlessly.

They've also adopted a very similar pricing model to the other services, with a free tier for up to five users, and an Essentials, Standard or Enterprise tier accommodating additional users/voice notifications accordingly.

Our score: 5/5



AWS

In terms of being a service provider, AWS covers much more than the features we are discussing here. But within that maze of services, there are many tools that can be utilised to carry out the functionality we are discussing. Much like most AWS services, the pricing model is more granular, so rather than a monthly subscription you pay for what you use.

Source control – CodeCommit

Much like the rest, CodeCommit is a Git repository service. It also supports some of the same features, with the key difference being in pricing. There is a free tier, for up to five users, with up to 50GB storage and up to 10,000 Git requests per month. Beyond this, it's \$1 per user per month.

The user interface is designed to match the wider AWS theme, and thus might appear to be less intuitive when compared with GitHub or Bitbucket, both designed solely with Git usage in mind. When the service was first released it supported considerably fewer features, but over time AWS have gradually extended it.

Our score: 3/5

Documentation – WorkDocs

WorkDocs is not a dedicated documentation service, but it is a platform for securely sharing and accessing documents across an organisation. It also supports version control for documents. It may not have the full feature set of a dedicated documentation service such as Confluence or Google Docs. Considering that its primary purpose is far more general than DevOps/documentation, it's inevitably not going to be as feature-rich as some of the other offerings.

It also costs \$5 per user per month (unless you're already an Amazon Workspaces customer) plus \$0.023 per TB per month above the 1TB free allowance.

Our score: 2/5

Pipelines – CodePipeline, CodeBuild

The Code* suite of services integrates heavily with many other AWS services. However, the principles are the same as many of the other CI/CD offerings.

CodePipeline allows you to define a pipeline starting with source control. It generally will be triggered on a commit to a particular branch. The artefacts from source control are stored in S3 and then used by anything else that is defined in the pipeline. There are six different types of action:

- Source
- Build
- Test
- Deploy
- Approval
- Invoke

Some possible actions in CodePipeline:

- Invoke a Lambda Function
- Start a Codebuild Project
- Deploy with CodeDeploy

CodePipeline is very dynamic with the options for creating a workflow. Regarding pricing, it's a simple pay-for-what-you-use model. \$1 per month for each active pipeline. Codebuild is similar to the other CI offerings we are comparing. You define what you want to do in a 'buildspec'. When the build is kicked off, a container is created, the source code pulled down, and whatever you have defined in the buildspec will be actioned before the container is killed. You are billed per minute based on how much compute/memory capacity you need.

CodeDeploy is also available to facilitate deployments whilst avoiding downtime.

Our score: 4/5

Package storage – CodeArtifact

Regardless of what language you're writing your applications in, CodeArtifact can store it for privately managed distribution. As with any AWS service, it has granular security features, as well as being highly scalable and highly available. The pricing model is simpler than many other AWS services – \$0.05 per GB per month for package storage and \$0.05 per 10,000 requests.

Our score: 5/5

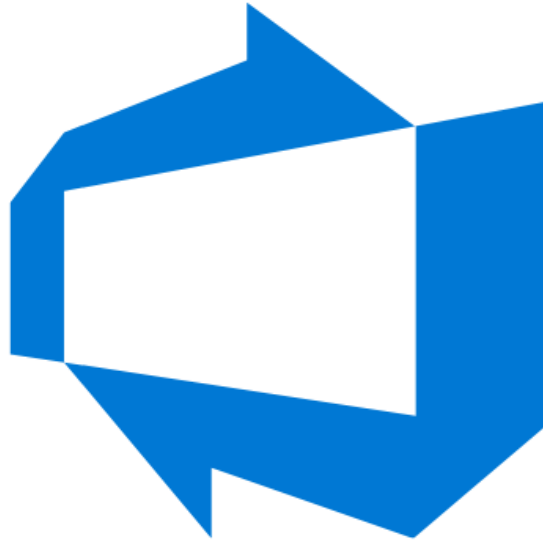
Incident management – CloudWatch/SNS

This is certainly not a dedicated incident management service, but a way to utilise two services. CloudWatch is a way of tracking metrics – AWS tracks lots of metrics out of the box but you can push any metric you want to CloudWatch for graphing/alerting.

If you define a particular threshold and the metric breaches that threshold, the alarm can trigger an SNS (Simple Notification Service) topic. From there, people can be alerted via email, SMS or a webhook to another third-party service.

Compared with the other incident management solutions we discussed, this probably has the steepest learning curve and will take the most time to implement. It also has a more complex pricing model as two different services are being utilised, each with its own pricing.

Our score: 3/5



Azure DevOps

Microsoft

Source control – Azure DevOps Services

Azure offers a robust source control solution called Azure DevOps Services (formerly known as Visual Studio Team Services or VSTS). Azure DevOps Services supports popular version control systems like Git and Team Foundation Version Control (TFVC) and integrates seamlessly with Azure services.

Our score: 4/5

Documentation – Microsoft Learn

Azure Microsoft Learn documentation is a comprehensive online resource provided by Microsoft to help individuals and teams learn about Azure cloud services and technologies. It offers a wide range of tutorials, hands-on labs, courses, and documentation to assist users in acquiring Azure-related skills and knowledge. Microsoft Learn's interactive and structured approach allows users to gain practical experience and expertise in areas such as cloud computing, AI, machine learning, and more, making it a valuable resource for Azure users.

One of the drawbacks of Microsoft Learn primarily revolves around its focus on Microsoft technologies and the need to supplement it with additional resources for a more comprehensive skill set. Depending on your learning goals and needs, these factors may or may not be significant concerns.

Our score: 3/5

Pipelines – Azure Pipelines

Azure Pipelines is a cloud-based continuous integration and continuous delivery (CI/CD) service provided by Microsoft Azure. It helps automate the building, testing, and deployment of software applications across various platforms and environments. Developers can define pipelines as code to specify how their code should be built, tested, and deployed, enabling a streamlined and automated software delivery process.

Azure Pipelines was initially developed as part of Visual Studio Team Services (VSTS), which was first announced in 2013. It has since evolved and became a key component of Azure DevOps Services when the Azure DevOps suite was rebranded from VSTS in September 2018. While the specific service name and branding changed in 2018, the development of the underlying technology began several years earlier.

Our score: 4/5

Task management and bug tracking – Azure Boards

Azure Boards is a work tracking system and project management tool provided as part of the Azure DevOps Services suite. It helps development teams plan, track, and manage their work items, including tasks, user stories, bugs, and features. Teams can use it to organise and prioritise work, visualise progress using customizable boards and dashboards, and foster collaboration among team members.

Azure Boards, part of Azure DevOps, evolved from Visual Studio Online in 2013, rebranded as Visual Studio Team Services (VSTS) in 2015, and became a core component of Azure DevOps Services in 2018.

Our score: 5/5

Package storage – Azure Artefacts

Azure Artefacts is a service in Azure DevOps that allows you to create, host, and share software packages and dependencies, facilitating efficient package management for your development projects. It was introduced as part of Azure DevOps Services in 2018, providing a centralised solution for managing software packages and artefacts within the Azure DevOps ecosystem.

Our score: 5/5

Incident management – Azure Security Centre

Azure Security Center is a Microsoft Azure service that helps safeguard your cloud resources by providing advanced threat protection across hybrid workloads, helping you detect and respond to security threats and vulnerabilities. It was introduced by Microsoft in 2015 as a security management system for Azure cloud environments.

Azure Security Center initially focused on providing security recommendations and threat detection for Azure resources. Over time, it evolved to support hybrid environments and extended its capabilities to include threat protection for on-premises and multi-cloud workloads. Azure Security Center continues to develop and adapt to meet the evolving security needs of organisations in the cloud and hybrid environments.

Our score: 4/5



GitHub

GitHub is predominantly a source control service but the product offering has evolved since its inception in 2008. In 2018 Microsoft acquired GitHub, and then in 2020, GitHub acquired npm (the NodeJS package repository).

Source control – GitHub

As mentioned above – and the clue is in the name – GitHub is primarily a Git source control system. It's about as feature-rich as any others but has the benefit of being arguably the widest adopted Git provider.

The standard offering is SaaS, supporting both public and private repositories. GitHub Enterprise has a self-managed offering if you need on-prem hosting. With the free tier, individuals can have unlimited public/private repositories. As soon as you need to unlock the collaboration features (branch protections, code owners, required reviewers) then the Team plan is \$4 per user per month.

Our score: 5/5

Documentation – README

This is not so much of a documentation service, as README is a common way to represent Markdown documentation in version control. But the way in which GitHub represents these files allows for such documents to be easily read and having the documentation within version control, alongside the code it is documenting, makes a lot of sense. There is also no additional cost to this, beyond your usual GitHub usage.

Our score: 4/5

Pipelines – GitHub Actions

A relatively new offering for GitHub, Actions follows the same process as Codebuild or Bitbucket Pipelines, in that containers are used to build/test your code.

Pricing is for the time it takes for workflows to run, with 2,000 minutes per month on the Free plan. The curve is not steep as the tiers increase, with the Teams plan only allowing 3,000 minutes per month. This is followed by a sharp increase to 50,000 minutes on the Enterprise plan. If you have many services, each running many workflows throughout the day, you may find that you run out of minutes, with a steep pricing increase being your next option.

Our score: 4/5

Task management – GitHub Projects

GitHub Projects was designed to integrate tightly with Issues (covered below), allowing teams to track changes in the codebase specifically. With this in mind, it perhaps doesn't work as well as Jira for any tasks that don't have an explicit requirement for code modification. It does have a set of analytics/reporting features but isn't quite as feature-rich or configurable as Jira (although your business may prefer this fact).

There is no additional cost to using Projects, beyond your usual GitHub usage.

Our score: 4/5

Bug tracking – GitHub Issues

As mentioned above, Issues is tightly integrated with Projects. For many open-source public repositories, Issues are the entry point for public feedback, issues, feature requests and any other interaction with users. The feedback loop is versatile, with conversations taking place within an Issue. Once this can be translated into actionable work within the codebase, it can be broken down into tasks, to be worked on and branches/commits/PRs can be associated with said tasks to allow for tracking, either within Issues or within Projects.

As per the other features, Issues can be used at no extra cost.

Our score: 4/5

Package storage – GitHub Packages

Packages is GitHub's answer to private artefact hosting. If you're using other GitHub services (e.g. Actions) then it would make more sense to use this for package storage than, say, CodeArtifact.

Packages are free for public repositories but your packages will then also be public. For private repositories, there is an allowance of 500MB storage on the free tier, with 2GB for Teams and 50GB for Enterprise. This pricing model isn't quite as dynamically scalable as CodeArtifact.

Our score: 4/5

Google Cloud Platform



Google

Source control – Git/GCP Source Repo

Google Cloud Platform (GCP) primarily uses Git for source code management and version control. Google offers Google Cloud Source Repositories, a service that integrates Git repositories directly with GCP. Users can use popular Git clients and tools to interact with their repositories in Google Cloud Source Repositories, making it a familiar and convenient choice for source control in GCP.

Our score: 4/5

Documentation – Cloud Docs

Google Cloud Platform's Cloud Docs offer a comprehensive resource for users, featuring structured learning paths, interactive examples, and real-world use cases to cater to users at all levels. With its powerful search and navigation capabilities, users can easily access versioned documentation and stay updated with GCP's evolving services. The platform's strong community engagement links users to relevant discussions and forums, fostering a collaborative environment.

Our score: 4/5

Pipelines – Cloud Build

Google Cloud Build is a managed continuous integration and continuous delivery (CI/CD) platform provided by Google Cloud Platform (GCP). It automates and streamlines the building, testing, and deployment of applications in the cloud. Developers can configure build pipelines using build triggers to automatically build and deploy code changes whenever new commits are pushed to a source code repository. Google Cloud Build supports integration with popular version control systems like Git, making it a valuable tool for creating efficient and automated software development workflows in GCP.

Google Cloud Build was introduced in 2018 as a CI/CD service within Google Cloud Platform, providing developers with a scalable and fully managed platform for automating the build and deployment of applications in the cloud.

Our score: 4/5

Package storage – Artefact Registry

Google Cloud Artefact Registry is a managed service that allows organisations to store, manage, and secure their software artefacts, such as Docker container images and Maven/Gradle packages. It provides a centralised and versioned repository for these artefacts, making it easier to manage dependencies and ensure consistency in software development and deployment pipelines. Google Cloud Artefact Registry also integrates with Google Cloud services like Google Kubernetes Engine (GKE) and Cloud Build, facilitating seamless artefact management and deployment within the Google Cloud ecosystem.

Our score: 5/5



Conclusions

An immediate benefit to all of these solutions is the level of integrations that they provide. We have deliberately not mentioned the third-party integrations for each provider as this guide would be far too long. Clearly, the intra-product integrations for each provider are very tight. If you don't mind a degree of vendor lock-in, picking one provider and using several of their services will certainly bring value. That said, some providers do have better integrations with third parties than others. For example, CodePipeline can be triggered from a number of different version control systems, whereas GitHub Actions requires the use of GitHub.

An interesting new product within GitHub that we haven't really covered here is Copilot – a generative AI coding assistant. As you write your code, it will suggest entire blocks based on a small amount of information that you give it. In practice, it won't replace developers, but when it is correct it can save time, allowing developers to focus their energy elsewhere.

Which tools should you choose?

Choosing the correct tools for your organisation depends on your priorities. If you want to keep as much of your configuration as possible within the codebase, then GitHub is probably your best bet. If you want your DevOps tools to be tightly integrated with your cloud compute provider, then either Microsoft, Google or AWS will be an obvious choice depending on where your application is hosted. If the focus of your tooling is on your Agile workflow, then Atlassian is probably the right fit for you.

Whilst we have compared these providers as all-in-one solutions, there is never a requirement to commit to vendor lock-in. Most of the services offered can be subscribed to individually. If you do have a specific requirement for a centralised logging solution only, for example, then something like Papertrail or Loggly may be more appropriate for your use case. Or if all you want is task management, a tool like Asana or Monday may suffice.

Whether or not you adopt a single-vendor or multi-vendor solution depends on your business priorities.

Benefits of single-vendor architecture

- **Streamlined integration:** With a single-vendor architecture, all your DevOps tools are designed to work seamlessly together, reducing integration complexities. This streamlined integration ensures smooth collaboration across the entire toolchain.
- **Enhanced support and accountability:** By dealing with only one vendor, you benefit from having a single point of contact for support and maintenance. This results in faster issue resolution and clear accountability, ensuring that your DevOps processes stay up and running without delays.
- **Simplified vendor management:** Dealing with multiple vendors can be time-consuming and administratively burdensome. However, with a single-vendor approach, you have the advantage of simplified vendor management, reducing administrative overhead and potential conflicts between different vendors.
- **Cohesiveness and consistency:** With a single-vendor architecture, you can ensure that all your DevOps tools are aligned, consistent, and compatible with each other. This cohesiveness helps create a unified and efficient DevOps environment.

Benefits of multi-vendor architecture

- **Flexibility and customisation:** Multi-vendor architecture allows you to choose best-of-breed solutions from different vendors, tailoring your toolchain to meet your specific needs. This flexibility enables you to leverage the most suitable tools for each stage of your DevOps pipeline.
- **Innovation and competitive advantage:** By adopting a multi-vendor approach, you gain access to a wider range of cutting-edge technologies and solutions. This can help your organisation stay ahead of your competition, as you can adopt the latest tools and practices in the industry.
- **Risk mitigation and scalability:** Relying on a single vendor for all your DevOps tools may introduce risks such as dependency and limited scalability. In contrast, multi-vendor architecture mitigates these risks by providing options for redundancy and scalability, giving you the freedom to scale your DevOps processes as needed.
- **Negotiating power:** Engaging with multiple vendors enhances your negotiation power. This allows you to secure competitive pricing, favourable contract terms, and improved service-level agreements, ultimately optimising your return on investment.

Have you got the tools you need? Our experts are here to help

With so many tools available, choosing the right ones for your organisation can be confusing.

Our expert DevOps consultants are here to help you adopt the tools you need to reap the rewards of effective DevOps implementation.

Speak with our team
and kickstart your
DevOps transformation
journey today

Contact Us:

T: +44 (0) 20 3130 4811

E: contact@ten10.com

W: www.ten10.com

