## What is AWS (Amazon Web Services)

Amazon Web Services (AWS) is a comprehensive cloud computing platform provided by Amazon. It offers a vast array of services, including computing power, storage solutions, and databases, empowering businesses to scale and innovate without the need for substantial upfront investments. Trusted by millions, AWS is known for its reliability, flexibility, and extensive global infrastructure. AWS boasts a vast portfolio of over 200 cloud services, offering an expansive range from compute and edge computing to serverless solutions. Key offerings include Amazon EC2 for computing, Amazon S3 for elastic object storage, Amazon EBS for networking, and Amazon Lambda for serverless computing.

AWS is an exceptional choice for those seeking a public cloud provider excelling in Infrastructure-as-a-Service (IaaS) with pay-as-you-go pricing and autoscaling resources. With IaaS, AWS simplifies the cloud experience, providing servers, networking, computing, and storage resources without the hassle of procuring or managing infrastructure. It’s the epitome of cloud flexibility and efficiency.

## What is Microsoft Azure

Microsoft Azure is a cloud computing platform that provides a wide range of services to help organizations build, manage, and deploy applications globally. With an emphasis on seamless integration with Microsoft’s ecosystem, Azure offers robust computing, analytics, networking, and more solutions. It’s a trusted choice for enterprises, offering a hybrid approach and strong support for diverse workloads.

A couple of standout features underline Azure’s popularity among enterprises. Firstly, the ease of transition is amplified for many enterprises already utilizing Microsoft products like Microsoft 365, SQL Server, SharePoint, and Power BI. Integrating Microsoft’s cloud services becomes a logical and efficient extension of their existing technology stack.

Moreover, Azure’s commitment to a hybrid cloud approach resonates well with modern enterprises. Integrating on-premises and public cloud services addresses the growing trend where companies desire to maintain their data centers while harnessing the benefits of the cloud. In this context, Azure emerges as a strategic ally, facilitating a seamless blend of on-premises infrastructure with the agility and innovation the cloud offers (see [on premise vs cloud](https://cloudfresh.com/en/blog/on-premise-vs-cloud/) as well).

## What is Google Cloud

Google Cloud is a suite of cloud computing services from Google designed to assist businesses in leveraging the power of data and innovation. With a focus on data analytics, machine learning, and open-source technologies, Google Cloud Platform (further, GCP) offers a flexible and scalable infrastructure. It’s a preferred choice for organizations seeking cutting-edge solutions and a strong foundation for digital transformation.

Google Cloud is renowned for its cutting-edge security options, setting industry standards for robust data and systems protection, and providing hybrid and multi-cloud solutions, offering unparalleled flexibility for organizations. GCP also provides top-notch data warehouse solutions, allowing organizations to store, manage, and analyze vast amounts of data efficiently. Its innovative Business Intelligence (BI) and Machine Learning (ML) features stand out. Businesses can leverage these tools to derive meaningful insights from their data, drive informed decision-making, and stay at the forefront of technological innovation.

## Amazon Web Services vs. Azure vs. Google Cloud

**History and Market Presence**

* [**Amazon Web Services (AWS)**](https://www.mgt-commerce.com/tutorial/magento-on-kubernetes-aws/) launched in 2006. It offered cloud computing to businesses as a pay-as-you-go service. It started with just a few basic services like storage and computation but quickly expanded.
* AWS is currently the leader in the cloud market. It holds a **33% market share**.
* **Microsoft Azure** was introduced in 2010, initially called Windows Azure. It is the second-largest cloud provider. It aims to provide cloud services for computing, analytics, storage, and networking.
* Microsoft Azure follows closely behind with a **24% market share** in cloud infrastructure.
* **Google Cloud Platform (GCP)** entered the market in 2011. While Google was running the search engine, its expansion was focused on large-scale computing solutions and data analytics.
* GCP holds an **11% market share**. It's popular among companies that rely on analytics and machine learning.

**Comparison of Core Services**

**1. AWS Compute power**

* **Elastic Compute Cloud (EC2)** is a service that allows businesses to run virtual servers in the cloud.
* You can select the amount of computing power and memory needed for your applications.
* EC2 can scale up or down automatically, depending on your application. For example, you might need more power during busy hours and less at other times.
* It's suitable for small to large web applications that handle millions of users.
* AWS also offers database services. Amazon RDS is used for relational databases, DynamoDB for NoSQL solutions, and Redshift for data warehousing.
* [**Amazon RDS**](https://www.mgt-commerce.com/magento-database-amazon-rds) allows you to launch a database with just an API call. You don’t need to worry about the underlying database setup.
* DynamoDB can handle more than 10 trillion requests/day and can support peaks of 20 million requests/second.
* Redshift allows you to share your data across Redshift clusters and with other AWS accounts.
* AWS offers a suite of developer tools like AWS CodeBuild, [**CodeDeploy**](https://www.mgt-commerce.com/magento-zero-downtime-deployment-mgt-code-deploy), and CodePipeline. These facilitate code integration and delivery.
* [**Amazon CloudFront**](https://www.mgt-commerce.com/best-cdn-for-magento-amazon-cloudfront) is a fast CDN service that securely delivers media, data, and APIs to customers globally. It is done with low latency and high transfer speeds.

**2. Azure Compute Power**

* **Virtual Machines (VMs)** in Azure are on-demand, [**scalable cloud**](https://www.mgt-commerce.com/blog/magento-hosting-providers-best-scalable-cloud-servers/) computing resources. They are similar to physical desktops but hosted in Azure's cloud.
* Azure VMs offer the ability to scale your computational resources up or down.
* Azure CDN leverages Microsoft's global network to deliver content efficiently. It offers strong integration with Azure services.
* If your business uses Azure storage solutions, Azure CDN can automatically [**cache**](https://www.mgt-commerce.com/mgt-varnish-cache-for-magento-performance) data and content from these services.
* **Azure HDInsight** is a cloud-based service that processes big data in real-time. It is used to analyze customer data and optimize operational decisions.
* **Azure Cosmos DB** provides a globally distributed, multi-model database service. It is used for managing data at scale.
* Azure offers developer tools like Visual Studio and Visual Studio Code integration. It makes it a favorite for Windows and .NET developers.

**3. Google Cloud Platform Compute Power**

* **Compute Engine** provides resizable compute capacity in the cloud. It allows you to spin up virtual machines as needed quickly.
* One of the standout features of Compute Engine is its ability to perform live migrations of Virtual Machines.
* If the hardware hosting your VM needs maintenance, Google can move your VM to another host without downtime.
* Google Cloud’s database services are Cloud SQL for relational databases, Firestore for NoSQL document storage, and BigQuery for data warehousing.
* GCP databases integrate well with **advanced analytics, Google AI, and machine learning** tools.
* Cloud SQL scales smoothly to store more data and handle more queries without managing the infrastructure.
* Firestore supports complex queries like filtering and sorting over multiple fields. It does this without lowering performance.
* BigQuery can process terabytes of data in seconds and petabytes in minutes. It integrates with Google’s [**data analytics**](https://www.mgt-commerce.com/blog/magento-analytics/) and machine learning tools.

All three major cloud providers, AWS, Azure, and GCP, offer robust services, making them top contenders in the cloud computing landscape. When choosing between Azure vs Google vs AWS, factors such as region availability, service ecosystem, hybrid cloud capabilities, computing, storage, analytics, and other services play crucial roles. Ultimately, the choice among AWS, Azure, and GCP depends on specific business needs, preferences, and existing technology stacks.

AWS stands out for its extensive service portfolio and widespread adoption, while Azure excels in seamless integration with Microsoft's ecosystem and a hybrid cloud approach. Although the smallest among the big three, Google Cloud shines with cutting-edge security, innovative BI, AI and ML features, and a strong focus on data analytics.

Google Cloud uniquely excels in several areas. It prioritizes best-in-class security, provides flexibility with hybrid cloud options, and stands out for user-friendly, fully managed services, especially in serverless computing. Google Cloud’s commitment to fostering a culture of innovation and openness to third-party solutions, following the hybrid and multi-cloud approach, and leveraging the best of Google’s technologies further sets it apart, making it a compelling choice for organizations seeking cutting-edge solutions. Also, GCP is providing an uneven instrument set for working with AI and ML tasks, including custom Google-developed TPU processors. While all three are strong contenders, Google Cloud’s distinctive strengths make it an attractive option for those prioritizing innovation, security, and flexibility.

If you decide to start or continue your cloud journey with the Google Cloud, we’re here. We will help you optimize your IT infrastructure, develop integrations for better system interoperability, and create new structures and processes for your teams. At the same time, our [support experts](https://cloudfresh.com/en/services/google/) will provide you with the best customer experience!

## BigQuery

BigQuery ML lets you [create and run machine learning (ML) models](https://cloud.google.com/bigquery/docs/e2e-journey) by using GoogleSQL queries. It also lets you access [Vertex AI models](https://cloud.google.com/bigquery/docs/generative-ai-overview) and [Cloud AI APIs](https://cloud.google.com/bigquery/docs/ai-application-overview) to perform artificial intelligence (AI) tasks like text generation or machine translation. Gemini for Google Cloud also provides AI-powered assistance for BigQuery tasks.

Usually, performing ML or AI on large datasets requires extensive programming and knowledge of ML frameworks. These requirements restrict solution development to a very small set of people within each company, and they exclude data analysts who understand the data but have limited ML knowledge and programming expertise. However, with BigQuery ML, SQL practitioners can use existing SQL tools and skills to build and evaluate models, and to generate results from LLMs and Cloud AI APIs.

You can work with BigQuery ML capabilities by using the following:

* The Google Cloud console
* The bq command-line tool
* The BigQuery REST API
* Integrated [Colab Enterprise notebooks in BigQuery](https://cloud.google.com/bigquery/docs/notebooks-introduction)
* External tools such as a Jupyter notebook or business intelligence platform

**Advantages of BigQuery ML**

BigQuery ML offers several advantages over other approaches to using ML or AI with a cloud-based data warehouse:

* BigQuery ML democratizes the use of ML and AI by empowering data analysts, the primary data warehouse users, to build and run models using existing business intelligence tools and spreadsheets. Predictive analytics can guide business decision-making across the organization.
* You don't need to program an ML or AI solution using Python or Java. You train models and access AI resources by using SQL—a language that's familiar to data analysts.
* BigQuery ML increases the speed of model development and innovation by removing the need to move data from the data warehouse. Instead, BigQuery ML brings ML to the data, which offers the following advantages:
  + Reduced complexity because fewer tools are required.
  + Increased speed to production because moving and formatting large amounts of data for Python-based ML frameworks isn't required to train a model in BigQuery.

For more information, watch the video [How to accelerate machine learning development with BigQuery ML](https://www.youtube.com/watch?v=EUPBVv9tp38).

**Generative AI and pretrained models**

You can use [remote models](https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-create-remote-model) to access Vertex AI models and Cloud AI APIs from BigQuery ML in order to perform the following tasks:

* [Generative AI](https://cloud.google.com/bigquery/docs/generative-ai-overview#generative_ai) by using Vertex AI text or multimodal [models](https://cloud.google.com/vertex-ai/docs/generative-ai/learn/models#foundation_models).
* [Text or multimodal embedding](https://cloud.google.com/bigquery/docs/generative-ai-overview#text_embedding) by using Vertex AI embedding models.
* [Natural language processing](https://cloud.google.com/bigquery/docs/ai-application-overview#natural_language_processing) by using the [Cloud Natural Language API](https://cloud.google.com/natural-language).
* [Machine translation](https://cloud.google.com/bigquery/docs/ai-application-overview#machine_translation) by using the [Cloud Translation API](https://cloud.google.com/translate).
* [Document processing](https://cloud.google.com/bigquery/docs/ai-application-overview#document_processing) by using the [Document AI API](https://cloud.google.com/document-ai).
* [Audio transcription](https://cloud.google.com/bigquery/docs/ai-application-overview#audio_transcription) by using the [Speech-to-Text API](https://cloud.google.com/speech-to-text).
* [Computer vision](https://cloud.google.com/bigquery/docs/ai-application-overview#computer_vision) by using the [Cloud Vision API](https://cloud.google.com/vision).

**Supported models**

A [model](https://developers.google.com/machine-learning/glossary/#model) in BigQuery ML represents what an ML system has learned from training data. The following sections describe the types of models that BigQuery ML supports.

**Internally trained models**

The following models are built in to BigQuery ML:

* [Linear regression](https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-create-glm) is for forecasting. For example, this model forecasts the sales of an item on a given day. Labels are real-valued, meaning they cannot be positive infinity or negative infinity or a NaN (Not a Number).
* [Logistic regression](https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-create-glm) is for the classification of two or more possible values such as whether an input is low-value, medium-value, or high-value. Labels can have up to 50 unique values.
* [K-means clustering](https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-create-kmeans) is for data segmentation. For example, this model identifies customer segments. K-means is an unsupervised learning technique, so model training doesn't require labels or split data for training or evaluation.
* [Matrix factorization](https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-create-matrix-factorization) is for creating product recommendation systems. You can create product recommendations using historical customer behavior, transactions, and product ratings, and then use those recommendations for personalized customer experiences.
* [Principal component analysis (PCA)](https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-create-pca) is the process of computing the principal components and using them to perform a change of basis on the data. It's commonly used for dimensionality reduction by projecting each data point onto only the first few principal components to obtain lower-dimensional data while preserving as much of the data's variation as possible.
* [Time series](https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-create-time-series) is for performing time series forecasts. You can use this feature to create millions of time series models and use them for forecasting. The model automatically handles anomalies, seasonality, and holidays.

You can perform a [dry run](https://cloud.google.com/bigquery/docs/running-queries#dry-run) on the CREATE MODEL statements for internally trained models to get an estimate of how much data they will process if you run them.

**Externally trained models**

The following models are external to BigQuery ML and trained in Vertex AI:

* [Deep neural network (DNN)](https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-create-dnn-models) is for creating TensorFlow-based deep neural networks for classification and regression models.
* [Wide & Deep](https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-create-wnd-models) is useful for generic large-scale regression and classification problems with sparse inputs ([categorical features](https://en.wikipedia.org/wiki/Categorical_variable) with a large number of possible feature values), such as recommender systems, search, and ranking problems.
* [Autoencoder](https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-create-autoencoder) is for creating TensorFlow-based models with the support of sparse data representations. You can use the models in BigQuery ML for tasks such as unsupervised anomaly detection and non-linear dimensionality reduction.
* [Boosted Tree](https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-create-boosted-tree) is for creating classification and regression models that are based on [XGBoost](https://xgboost.readthedocs.io/en/latest/).
* [Random forest](https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-create-random-forest) is for constructing multiple learning method decision trees for classification, regression, and other tasks at training time.
* [AutoML](https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-create-automl) is a supervised ML service that builds and deploys classification and regression models on tabular data at high speed and scale.

You can't perform a [dry run](https://cloud.google.com/bigquery/docs/running-queries#dry-run) on the CREATE MODEL statements for externally trained models to get an estimate of how much data they will process if you run them.

**Remote models**

You can create [remote models](https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-create-remote-model#endpoint) in BigQuery that use models deployed to [Vertex AI](https://cloud.google.com/vertex-ai/docs). You reference the deployed model by specifying the model's [HTTPS endpoint](https://cloud.google.com/vertex-ai/docs/general/deployment#what_happens_when_you_deploy_a_model) in the remote model's CREATE MODEL statement.

The CREATE MODEL statements for remote models don't process any bytes and don't incur BigQuery charges.

**Imported models**

BigQuery ML lets you import custom models that are trained outside of BigQuery and then perform prediction within BigQuery. You can import the following models into BigQuery from [Cloud Storage](https://cloud.google.com/storage):

* [Open Neural Network Exchange (ONNX)](https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-create-onnx) is an open standard format for representing ML models. Using ONNX, you can make models that are trained with popular ML frameworks like PyTorch and scikit-learn available in BigQuery ML.
* [TensorFlow](https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-create-tensorflow) is a free, open source software library for ML and artificial intelligence. You can use TensorFlow across a range of tasks, but it has a particular focus on training and inference of deep neural networks. You can load previously trained TensorFlow models into BigQuery as BigQuery ML models and then perform prediction in BigQuery ML.
* [TensorFlow Lite](https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-create-tflite) is a light version of TensorFlow for deployment on mobile devices, microcontrollers, and other edge devices. TensorFlow optimizes existing TensorFlow models for reduced model size and faster inference.
* [XGBoost](https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-create-xgboost) is an optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements ML algorithms under the [gradient boosting](https://en.wikipedia.org/wiki/Gradient_boosting) framework.

The CREATE MODEL statements for imported models don't process any bytes and don't incur BigQuery charges.

In BigQuery ML, you can use a model with data from multiple BigQuery Datasets for training and for prediction.

**Query syntax:** [**https://cloud.google.com/bigquery/docs/reference/standard-sql/query-syntax**](https://cloud.google.com/bigquery/docs/reference/standard-sql/query-syntax)

## What is Terraform?

HashiCorp Terraform is an infrastructure as code tool that lets you define both cloud and on-prem resources in human-readable configuration files that you can version, reuse, and share. You can then use a consistent workflow to provision and manage all of your infrastructure throughout its lifecycle. Terraform can manage low-level components like compute, storage, and networking resources, as well as high-level components like DNS entries and SaaS features.

**How does Terraform work?**

Terraform creates and manages resources on cloud platforms and other services through their application programming interfaces (APIs). Providers enable Terraform to work with virtually any platform or service with an accessible API.

HashiCorp and the Terraform community have already written **thousands of providers** to manage many different types of resources and services. You can find all publicly available providers on the [Terraform Registry](https://registry.terraform.io/), including Amazon Web Services (AWS), Azure, Google Cloud Platform (GCP), Kubernetes, Helm, GitHub, Splunk, DataDog, and many more.

The core Terraform workflow consists of three stages:

* **Write:** You define resources, which may be across multiple cloud providers and services. For example, you might create a configuration to deploy an application on virtual machines in a Virtual Private Cloud (VPC) network with security groups and a load balancer.
* **Plan:** Terraform creates an execution plan describing the infrastructure it will create, update, or destroy based on the existing infrastructure and your configuration.
* **Apply:** On approval, Terraform performs the proposed operations in the correct order, respecting any resource dependencies. For example, if you update the properties of a VPC and change the number of virtual machines in that VPC, Terraform will recreate the VPC before scaling the virtual machines.

## What is continuous integration (CI)?

[Continuous integration](https://about.gitlab.com/topics/ci-cd/benefits-continuous-integration/) is the practice of integrating all your code changes into the main branch of a shared source code repository early and often, automatically testing each change when you commit or merge them, and automatically kicking off a build. With continuous integration, errors and security issues can be identified and fixed more easily, and much earlier in the development process.

By merging changes frequently and triggering automatic testing and validation processes, you minimize the possibility of code conflict, even with multiple developers working on the same application. A secondary advantage is that you don't have to wait long for answers and can, if necessary, fix bugs and security issues while the topic is still fresh in your mind.

Common code validation processes start with a static code analysis that verifies the quality of the code. Once the code passes the static tests, automated CI routines package and compile the code for further automated testing. CI processes should have a version control system that tracks changes so you know the version of the code used.

## What is continuous delivery (CD)?

Continuous delivery is a software development practice that works in conjunction with CI to automate the infrastructure provisioning and application release process.

Once code has been tested and built as part of the CI process, CD takes over during the final stages to ensure it's packaged with everything it needs to deploy to any environment at any time. CD can cover everything from provisioning the infrastructure to deploying the application to the testing or production environment.

With CD, the software is built so that it can be deployed to production at any time. Then you can trigger the deployments manually or move to continuous deployment, where deployments are automated as well.

## What are CI/CD pipelines?

A [CI/CD pipeline](https://about.gitlab.com/topics/ci-cd/cicd-pipeline/) is an automated process utilized by software development teams to streamline the creation, testing and deployment of applications. "CI" represents continuous integration, where developers frequently merge code changes into a central repository, allowing early detection of issues. "CD" refers to continuous deployment or continuous delivery, which automates the application's release to its intended environment, ensuring that it is readily available to users. This pipeline is vital for teams aiming to improve software quality and speed up delivery through regular, reliable updates.

Integrating a CI/CD pipeline into your workflow significantly reduces the risk of errors in the deployment process. Automating builds and tests ensures that bugs are caught early and fixed promptly, maintaining high-quality software.

## What is meant by continuous testing?

Continuous testing is a software testing practice where tests are continuously run in order to identify bugs as soon as they are introduced into the codebase. In a CI/CD pipeline, continuous testing is typically performed automatically, with each code change triggering a series of tests to ensure that the application is still working as expected. This can help to identify problems early in the development process and prevent them from becoming more difficult and costly to fix later on. Continuous testing can also provide valuable feedback to developers about the quality of their code, helping them to identify and address potential issues before they are released to production.

In continuous testing, various types of tests are performed within the CI/CD pipeline. These can include:

* **Unit testing**, which checks that individual units of code work as expected
* **Integration testing**, which verifies how different modules or services within an application work together
* **Regression testing**, which is performed after a bug is fixed to ensure that specific bug won't occur again