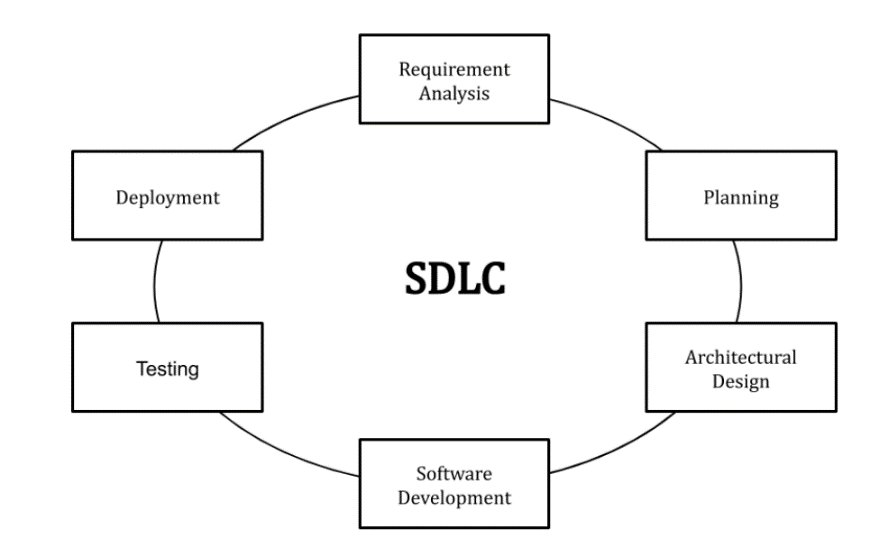
**SDLC :**

****

Stages and Best Practices

Following the best practices and/or stages of SDLC ensures the process works in a smooth, efficient, and productive way.

1. Identify the Current Problems

“What are the current problems?” This stage of the SDLC means getting input from all stakeholders, including customers, salespeople, industry experts, and programmers. Learn the strengths and weaknesses of the current system with improvement as the goal.

2. Plan

“What do we want?” In this stage of the SDLC, the team determines the cost and resources required for implementing the analyzed requirements. It also details the risks involved and provides sub-plans for softening those risks.

In other words, the team should determine the feasibility of the project and how they can implement the project successfully with the lowest risk in mind.

3. Design

“How will we get what we want?” This phase of the SDLC starts by turning the software specifications into a design plan called the Design Specification. All stakeholders then review this plan and offer feedback and suggestions. It’s crucial to have a plan for collecting and incorporating stakeholder input into this document. Failure at this stage will almost certainly result in cost overruns at best and the total collapse of the project at worst.

4. Build

“Let’s create what we want.”

At this stage, the actual development starts. It’s important that every developer sticks to the agreed blueprint. Also, make sure you have proper guidelines in place about the code style and practices.

For example, define a nomenclature for files or define a variable naming style such as camelCase. This will help your team to produce organized and consistent code that is easier to understand but also to test during the next phase.

5. Code Test

“Did we get what we want?” In this stage, we test for defects and deficiencies. We fix those issues until the product meets the original specifications.

In short, we want to verify if the code meets the defined requirements.

Try Stackify’s free code profiler, Prefix, to write better code on your workstation. Prefix works with .NET, Java, PHP, Node.js, Ruby, and Python.

6. Software Deployment

“Let’s start using what we got.”

At this stage, the goal is to deploy the software to the production environment so users can start using the product. However, many organizations choose to move the product through different deployment environments such as a testing or staging environment.This allows any stakeholders to safely play with the product before releasing it to the market. Besides, this allows any final mistakes to be caught before releasing the product.

The most common SDLC examples or SDLC models are listed below.

Waterfall Model

This SDLC model is the oldest and most straightforward. With this methodology, we finish one phase and then start the next. Each phase has its own mini-plan and each phase “waterfalls” into the next. The biggest drawback of this model is that small details left incomplete can hold up the entire process.

Agile Model

The Agile SDLC model separates the product into cycles and delivers a working product very quickly. This methodology produces a succession of releases. Testing of each release feeds back info that’s incorporated into the next version. [According to Robert Half](https://www.roberthalf.com/technology/blog/6-basic-sdlc-methodologies-the-pros-and-cons), the drawback of this model is that the heavy emphasis on customer interaction can lead the project in the wrong direction in some cases.

Iterative Model

This SDLC model emphasizes repetition. Developers create a version very quickly and for relatively little cost, then test and improve it through rapid and successive versions. One big disadvantage here is that it can eat up resources fast if left unchecked.

V-Shaped Model

An extension of the waterfall model, this SDLC methodology tests at each stage of development. As with waterfall, this process can run into roadblocks.

Big Bang Model

This high-risk SDLC model throws most of its resources at development and works best for small projects. It lacks the thorough requirements definition stage of the other methods.\

Spiral Model

The most flexible of the SDLC models, the spiral model is similar to the iterative model in its emphasis on repetition. The spiral model goes through the planning, design, build and test [phases](https://stackify.com/sdlc-phases-identify-problems/) over and over, with gradual improvements at each pass.

Benefits of the SDLC

SDLC done right can allow the highest level of management control and documentation. Developers understand what they should build and why. All parties agree on the goal upfront and see a clear plan for arriving at that goal. Everyone understands the costs and resources required.

AGILE and Scrum

 Scrum is the most widely used Agile framework and is based upon five values: commitment. courage, focus, openness, and respect. Scrum maintains many of the same roles as the basic Agile framework but adds the Scrum Master, who ensures Scrum is understood and executed properly.

Sprints – Sprints are timeboxes for accomplishing a goal. These remain consistent throughout the development process and their length will not exceed a single month.

Sprint planning – Sprint planning is the collaborative process of building the upcoming Sprint with the rest of the development team.

Daily Scrum – 15-minute meetings that occur during every day of the Sprint are called daily scrums. Achievements from the previous day are noted during daily scrums and new expectations are set in place.

Sprint review – These meetings occur at the end of every Sprint so the scrum team can present their Increment to stakeholders and receive feedback.

Sprint retrospective – At the end of a Sprint, teams meet to discuss the previous Sprint’s achievements and fallbacks before setting expectations and improvements for the next Sprint.

CLOUD

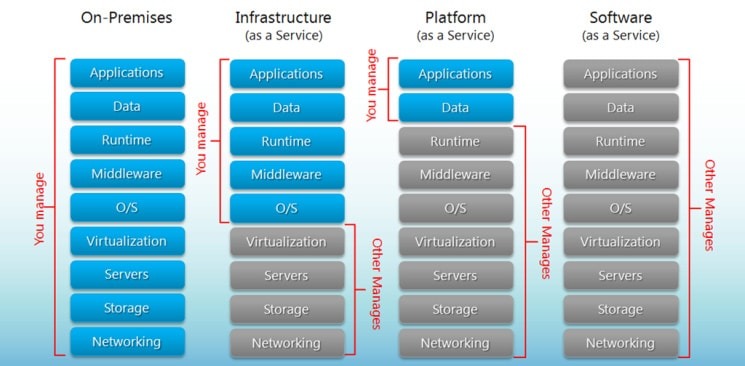
SaaS, PaaS, and IaaS are simply three ways to describe how you can use the cloud for your business.

• IaaS: cloud-based services, pay-as-you-go for services such as storage, networking, and virtualization.

• PaaS: hardware and software tools available over the internet.

• SaaS: software that’s available via a third-party over the internet.

• On-premise: software that’s installed in the same building as your business.



Sample: pizza as a service.

**Some of the Cloud providers:**

Amazon Web Services.

Microsoft Azure.

Google Cloud Platform.

Alibaba Cloud.

IBM.

Dell Technologies/VMware.

Hewlett Packard Enterprise

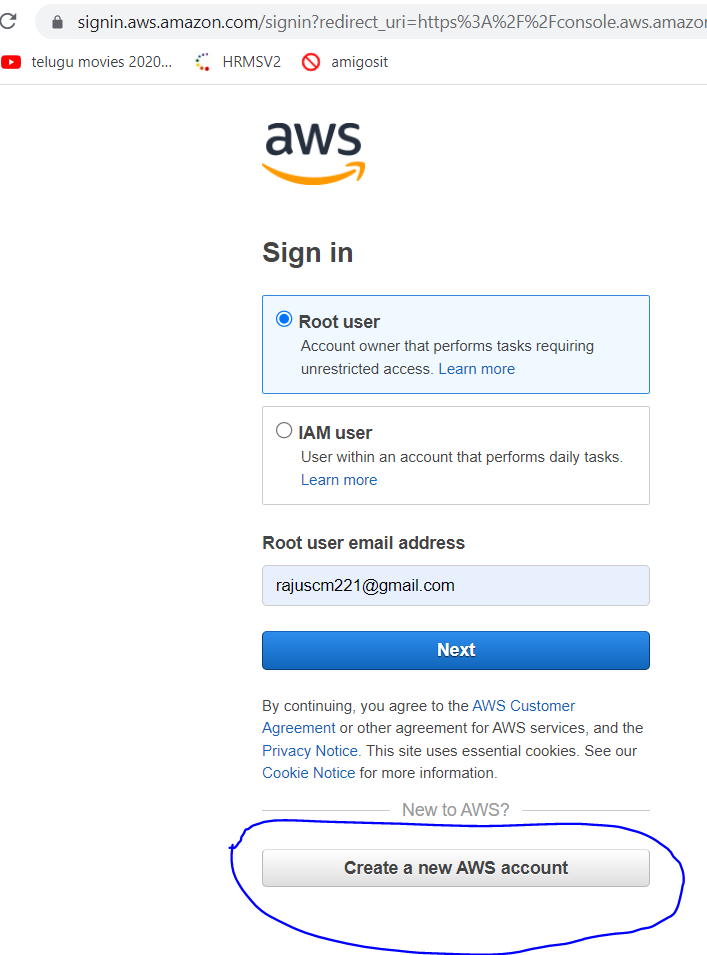
Cisco Systems.

In the third quarter of 2021, the most popular vendor in the cloud infrastructure services market, Amazon Web Services (AWS), controlled 32 percent of the entire market. Microsoft Azure takes second place with 21 percent market share, followed by Google Cloud with eight percent market share.

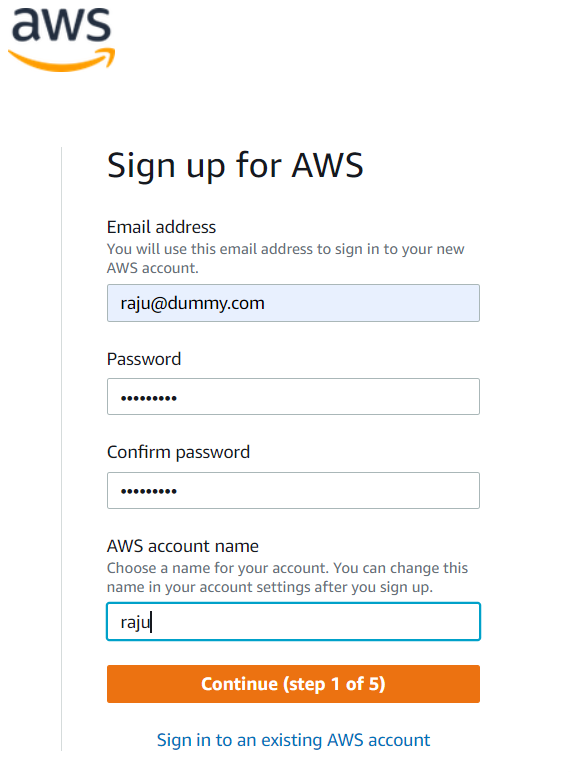
**AWS**

Account creation:

1. Open the [Amazon Web Services (AWS) home page](https://aws.amazon.com/).
2. Choose Create an AWS Account.  
   Note: If you signed in to AWS recently, choose Sign in to the Console. If Create a new AWS account isn't visible, first choose Sign in to a different account, and then choose Create a new AWS account.

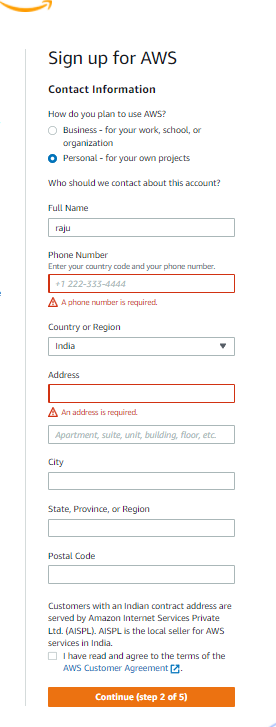


3.Enter your email address, password, AWS account name, and then choose Continue.



Add your contact information

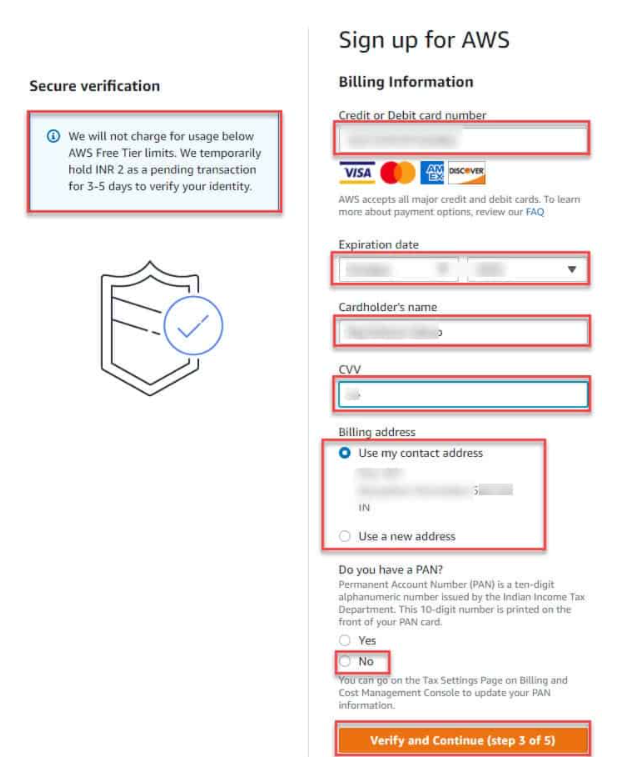
1. Select Business or Personal.  
   Note: Personal accounts and business accounts have the same features and functions.
2. Enter your company or personal information.  
   Important: For business AWS accounts, it's a best practice to enter the company phone number rather than a personal cell phone. Configuring a [root account](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_root-user.html) with an individual email address or a personal phone number can make your account insecure.
3. Read and accept the [AWS Customer Agreement](https://aws.amazon.com/agreement/).
4. Choose Continue.



### Add a payment method

On the Billing information page, enter the information about your payment method.

If you are signing up in India for an [Amazon Internet Services Private Limited (AISPL)](https://aws.amazon.com/premiumsupport/knowledge-center/what-is-aispl/) account, you must provide your CVV as part of the verification process. You might also have to enter a one-time password, depending on your bank. AISPL charges your payment method 2 Indian Rupees (INR), as part of the verification process. AISPL refunds the 2 INR after the verification is completed.



1.Now is the time to verify your Phone on the Confirm your Identity section. Provide the below details.

* How should we send you the verification code?: Select the text message radio button. You can also choose the Phone call option.
* Country or Region Code: Select your country or region code.
* Cell Phone Number: Provide the number of your cell phone.
* Security Check: Type the Exact captcha.

Click on the Send SMS button to receive the SMS on your mobile.

2. Enter the code you have received and then click on the Verify Code button.

3. It will show you now that “Your identity has been verified successfully.” Then click on the Continue button.

4. Now, on the next window you will see three plans. Choose Basic plan.

* Basic Plan (Free)
* Developer Plan
* Business Plan

**Networking**:

https://zhenye-na.github.io/aws-certs-cheatsheet/posts/vpc/