Full Stack Development

Day 1 L-3 [L:Lecture -3 : 3 Hours] FN-Fore Noon

1.What is an Enterprise?

Enterprise refers to a for-profit business started and run by an entrepreneur. And we will often say that people running such businesses are enterprising. A **business enterprise** consists of producing goods or services in exchange for commercial and financial benefits.

Examples of business enterprises include all the companies you pay to receive a good or service from. These may include your local shop or your Netflix subscription, both of which are business enterprises.

2.Organizing the Enterprise – process

The major difference between organization and enterprise is that the enterprise is profit-oriented while the organization is goal-oriented. The organization aims to achieve the shared objectives of the group or community, while the enterprise aims to generate revenues by addressing the needs of customers. The success of an enterprise depends on how efficiently the work environment is organized. An enterprise often comprises several organizations, also called teams, working together to achieve a common objective. Organizing the enterprise is the process of establishing relationships, delegating responsibilities, and defining authorities to manage the groups to work efficiently at the workplace.

The process of organizing comprises five steps. They are:

- Identify the work
- Group the work
- Establish the hierarchy
- Delegate the authority
- Coordinate

A properly organized process results in a workplace where team members have a clear understanding of their responsibilities. Organizing helps businesses reduce redundancies, avoid miscommunication and eliminate waste. Improper organization results in frustration, confusion and loss of efficiency among the workforce.

3 Understanding Types of Business Activities :

The business activities are classified into three different types namely operating activity, investing activity, and financing activity. Let us discuss the three types of business activities briefly:

1.Operating Activities: Operating activities refer to all those business activities that are directly or indirectly related to the provision of goods and services. As such they have a direct impact on cash flow, and eventually on income.

2.Investing Activities: Investing activities refers to all those activities that aimed to be capitalized for more than a year. This includes capital expenditure such as purchase of long term assets or real estate.

3.Financing Activities: Financing activities refer to all those activities that fund the business but are not directly related to the revenues from goods and services. Common financing activities include bonds, loans, and share issues.

4.What is business process?

- A business process or business method is a collection of related, structured activities or tasks by people or equipment which in a specific sequence produce a service or product (serves a particular business goal) for a particular customer or customers. In human language, business process is like an assembly line – it takes all the inputs in various stages and creates a fully serviceable finished product.
- Essentially anything that happens within your company that has a clearly defined start and finish can be considered a business process. Do you remember the hassle you went through when applying for a job at your (dream) employer? You have been a part of an HR business process that began with your application and resulted in the on-boarding week! How about that invoice that your customer receives after he made a purchase with your company? This is your typical order-to-cash process.

What are the key components of a business process?

For business process to be functional and sustainable, it must be in accordance with the following 8 key components:

1. **Have clearly defined boundaries** – you must know what goes in and what should come out
2. **Determine your structure** – in an ideal scenario any business process should follow a clear order
3. **Know your customer** – if you follow a business process, you must know who is at the receiving end and serve him
4. **Bring evident value** – if you serve your customers using business processes, it must bring them value and make their life easier
5. **Let the process become a part of your organization** – your organization cannot live without your process and vice versa
6. **Make your process universal** – if you plan thoroughly, one process can serve not one, but multiple different customers
7. **Avoid silos at all cost** – you don't want your backend developers to follow a process that only they know and can explain
8. **Identify the process owner** – someone must oversee the project and take full responsibility for its performance and improvement.

Business process categories and examples

Due to the nature of the business processes, it is quite possible to break up them into couple of distinct types, some will be vital for your core business, others will support it.

1.Operational processes

Operational processes, also called the core business processes, are the activities that bring direct value to your company and your customers. Such processes are absolutely essential to your company and its perfection is crucial because they are generating your revenue.
Examples of operational processes:

- order to cash process
- manufacturing of the products
- robot process automation (RPA)
- delivery of the products

2.Supporting processes

To be able to only focus on core business, activities must be every business owner's dream. As your organization grows, so do its business processes. In the real world, for every employee that focuses on core business and creates the value that consumers crave there are multiple workers that are not necessarily responsible for the final product but are helping to smooth the whole process. Without them, even the best organizations would collapse.

Examples of supporting processes:

- technical support
- accounting
- call center
- sales
- HR
- marketing funnels

Without these people and their respective processes, there is no future for any company.

3.Management processes

Somebody needs to oversee how the company runs. The managers surely have their set of rules and processes which they follow.

Their main task is to measure, monitor and control all the business-related procedures and systems.

Examples of management processes:

- budgeting
- governance (to oblige with the legal regulations and internal guidelines)
- infrastructure

5.Why to automate business process? (P-1) P-Practical -1 Hour

The future is automated and if you don't adapt, you'll be left in the dust. This fact is particularly true in the business world. Tooled with powerful Artificial Intelligence (AI), you can stay ahead of the curve and automate your small business or established company. When used properly, this AI can improve everything from productivity to overhead costs, and even the smallest companies can take advantage of this new technology.

[What Is Artificial Intelligence?

AI stands for Artificial Intelligence and it is central to the rapid automation of businesses both large and small. Artificial intelligence is a branch of computer science that studies the building of smart machines that are capable of performing tasks typically done by humans through the use of machine learning.]

Advantages or benefits of automated business process:

a. **Reduced Overhead Costs**

Profitability is a top priority for any business owner. Fortunately, automation can reduce one's operational costs and boost profits, by providing smart technologies that decrease the human errors.

On top of that, automated systems can reduce the time it takes to fulfil a task. For example, QA automation tools can accelerate processes .

b. **Improve Productivity**

According to a study performed in 2017 on marketing automation, automation technology increased sales productivity by 14.5%. This same study predicted that the desire for automation would only increase, and has grown by 30% every year in the United States.

c. **Streamline Communications**

Communication is key to any successful business, and with the proper automation tools, you can take your team's communications to the next level. Discussions carried out in emails, phone calls, or even text messages, can make for messy connections and lead to miscommunications.

d. **Increase Customer Satisfaction**

Regardless of your specific industry, being able to provide high-quality products and services faster and better will inevitably lead to increased customer satisfaction. That being said, there is also automation software out there that can improve the customer experience.

e. **Gain a Competitive Edge**

Based on a survey performed by The Economist Intelligence Unit, over 90% of organisations are already using technology to automate their business processes, and that percentage is only going to increase. Automation is here, and by jumping on it now, you can gain an edge against your competition.

Day 1 [L-1] L :Lecture -1 : 1 Hour AN [After Noon]

Digital transformation through Convergence of IT & OT

What is IT-OT convergence?

IT-OT convergence is a process of aligning Information Technology (IT) and Operational Technology (OT) enables processes to be combined and information to be shared across the two 'worlds'. This can be used to drive better management of systems and assets within organisation.

Previously the two systems have been independent – IT processes have largely revolved around data, information, communications-oriented systems and business networks, while OT is used to change, monitor or control physical devices, processes and events.

IT-OT convergence involves transferring information across both the IT and OT networks. This connectivity can be used to enhance the value the systems deliver and progress your organization further towards your business goal.

While many businesses want fully connected operations to get clear data from both sides, they can underestimate the planning needed to collect and correlate the right data, in the right format.

. By collecting the most valuable operational data from OT systems, your IT systems can process and analyse this data to generate the helpful insights needs that can be used to support real-time decision-making, which presents a huge opportunity for improving efficiency.

What other IT-OT convergence benefits are there?

- Sharing data previously held in disparate systems, resulting in information being more accessible to all parts of your business
- Improved visibility across the business, being able to analyse data as a whole instead of in isolation
- Reduced risk in manufacturing.
- Better disaster recovery processes - such as improved backup processes, or the use of cloud infrastructure with additional redundancy and automated failover in the event of a disaster.

True IT/OT convergence removes the gap between operational processes and information systems and combines them in a unified network architecture.

"Organisations mapping out their digital transformation journey may already understand the value of having systems that 'talk' to one another. Planning for integration may even be in the discussion as more organisations aim to take advantage of real-time data."

Bringing together disparate systems and breaking down the silos between departments provides opportunities to scale digital transformation projects (and business growth) while bringing greater value to both the organisation and customers.

Integrating your IT and OT systems can provide an intuitive, real-time representation of the current operational situation using modern technologies and standards. This includes the ability to **represent, trend, predict and resource accurately and automatically in real-tim**e.

What are the critical enablers for the convergence of OT and IT?

1. People who can work in cross-functional teams :

For successful IT-OT integration, you need the knowledge and support of both your operations and IT teams.A cross-functional team can explore potential challenges to IT-OT convergence within your business.

2. An IT-OT convergence strategy (and wider digital transformation strategy)

If your company is not clear on the value you hope to get from integrating IT and OT departments, it's crucial you take the time to determine this. Identify the priorities, and plan how to align systems to capability and your overall business strategy.

3. An understanding of your technology

An early step in the convergence process is understanding your current network state and the potential vulnerabilities.

Air gapped networks – the OT network is independent of IT networks, so there is currently no exchange of data or convergence. The computers or networks are not connected to the internet.

Interconnected networks - used when you don't want to join these networks. Instead, they are segregated to limit the data flowing back and forth – so only the data you want is being captured.

Either way, your IT and OT teams should be protecting each other's networks from different threats and attack vectors. Keep in mind that OT technology isn't as fast-moving as IT, so you need to identify who will be responsible for monitoring those networks.

Day 1 [P-2] P :Practical -2 : 2 Hours AN [After Noon]

Digital transformation through IT/OT convergence :

Case Study :

A copper mining company brought information and operational technology under a single governance and operating model to aid digital transformation.

Challenge :

With the goal of becoming a leader in the use of automation in its operations, a large copper mining company wanted to take advantage of an information technology (IT) and operational technology (OT) convergence movement sweeping across industries.

In a highly competitive and rapidly changing technology landscape, the company knew that a digital transformation, aimed at bringing IT and OT together under a common governance structure, could be the difference between surviving and thriving.

With Accenture's help, the company would launch a program to make better use of its technology and data, change the way it worked and ultimately, build a foundation to support its overall vision for digital transformation.

What Accenture did ?

The company and Accenture team designed and implemented unified technology governance and a common technology operating model across various sites and brought the management of IT and OT together under one new centralized technology organization.

Because of the complexity involved and because processes and technologies varied widely at different sites, the effort was divided into three phases:

1.Assessment:

The team used a Kanban board (a key tool to depict workflow visualization), created sticky notes to record ideas and gave presentations.

2.Design:

The team designed a global IT/OT convergence strategy, along with transition plans for each asset that would vary depending on the complexity.

3.Execution:

The team launched strategy and transition plans as the company moved to a new technology organization that encompassed both IT and OT.

People and culture

Because the effort involved a significant change in culture, stakeholders were consulted throughout the project, helping people understand what was happening and building buy-in for the new approach.

The company and Accenture also highlighted the fact that the convergence of IT and OT would benefit the workforce by creating a wider variety of technical career paths, opportunities to learn and develop skills, and the potential for employees to apply their skills to a broader range of roles across the company's operations and sites. Furthermore, they showed how the breaking down of silos and the integration of IT and OT teams will bring a more coordinated response to business requirements, the ability to share resources and exploit the same contracts, and significant cost reductions.

Value delivered

This initiative has provided greater visibility across the company's technology landscape and enabled IT and OT to operate under a single model, using higher-quality operational data.

The new approach allows management to optimize operations from a holistic perspective and use technology more efficiently and effectively. It has also allowed the company to enhance its focus on safety, production volume and operational costs. And it has enabled leading practices such as predictive asset management and integrated planning and scheduling.

Finally, the unified approach to IT and OT has positioned the company to continue to better take advantage of digital technologies. Technology professionals can work as an integrated team to identify and address IT/OT-related problems and move quickly to replicate improvements and innovations across the company—which will be key to realizing the company's transformation vision in the coming years.

Day 4 : Design Thinking L-1 Hr P-6 Hrs

4.1 What does design thinking exactly mean?

It is said to be "a formal method for practical, creative resolution of problems or issues, with the intent of an improved future result". Design thinking is a set of tools that enables solving a particular problem using analytics and creativity. Solving the problem is at core of design thinking. It is a human-centric approach that puts a customer in the center of attention. This approach is aimed at learning which product or service a user wishes for, exploring the ways to make it, developing and testing the product. It's about being solution oriented and focusing on the ways to solve the problem, not the problem itself.

Design thinking approach involves looking at already existing product and analyzing the ways to make it better and more user-friendly. It's about generating a plenty of new ideas, while placing the needs of customers above all. This approach can be applied everywhere, including business and non-business sectors.

4.2 The Relevance of Design in Software Development

Software development doesn't work just by itself. It has to empathize with the customer. Customers don't concentrate on which technologies were used while making the application or solution they need. They mainly focus on utility, which is why the developers also have to focus on it.

There is a common situation of misunderstanding happening between the solution developers and the customers. Of course, it can be said that this is a problem of miscommunication which can be solved by arranging more calls, meetings, and discussions. However, the solution may be found much deeper. Only understanding and empathizing with your clients can show you what they really want. And that is what you should start with.

So, it is not about the right techniques, but more about critical thinking. Nowadays, software development is everywhere. Nevertheless, with all the codes and technical tools everything is made by people for people. Software development is supposed to make our lives better and easier, which is why it's so important to understand what user needs and wants, and implement design thinking approach.

Design Thinking Process

The Institute of Design at Stanford developed the following 5 stages of design thinking process: empathy, defining, ideating, prototyping, and testing.

i.Empathy: Empathy is the ability to feel and understand what other people experience, It is an initial step of the entire design thinking process because empathizing with the customers enables understanding their needs and finding out what they really want, looking at the products and services from your users' perspective.

This goal can be achieved by the means of different interviews and observations. Interviewing must resemble conversation. You don't need to influence the answers because your main point is to understand a user. So, questions beginning with "Why" or asking the stories can give you valid results. However, what people tell can sometimes differ from what they do. That's why, it is so important to observe the interactions and notice what is easy, causes difficulties, and what can be improved.

ii. Define : After empathizing with your customers, there comes the phase of identification of the problem we need to solve. This step is about summing up and making sense of the information w received from the previous one. We have to setour problem statement and make it clear.

On this stage we have to consider thoroughly everything gathered while empathizing. We have to define a point of view, consisting the elements: users, their needs, and insights. We have to establish deep understanding the user. Outline a set of needs of user and provide the insights, based on your research and discoveries. Our goal is to set a frame, which makes the problem we are working on clearly defined and understandable, and encourages the members of our team to find the solutions.

iii. Ideate : Now, we need ideas. What we need is a plenty of creative solutions, multiple perspectives, and different viewpoints. It is an exploration phase, where all the ideas must be encouraged even if they seem to be not relevant enough. We have to imagine what we want to reach and all the ways to get there.

iv. Prototype : Then it's time to make a draft. You have to represent your idea in a clear and tangible form so that others could understand. Remember, it doesn't have to be perfect. It can be in a form of product, model or a sketch, but the primary aim of building a prototype is to pass our idea to the others and figure out howour product may feel and look.

v. Test : On this final stage of the design thinking process, your prototype is put into action and tested by people. You give it to users, see how they interact with it, how they implement it, and get your feedback. This phase allows you to understand what is right and what is wrong, and refine your ideas.

For Practical Sessions of 6 Hrs duration ,study the concept of Design Thinking through the following Videos.

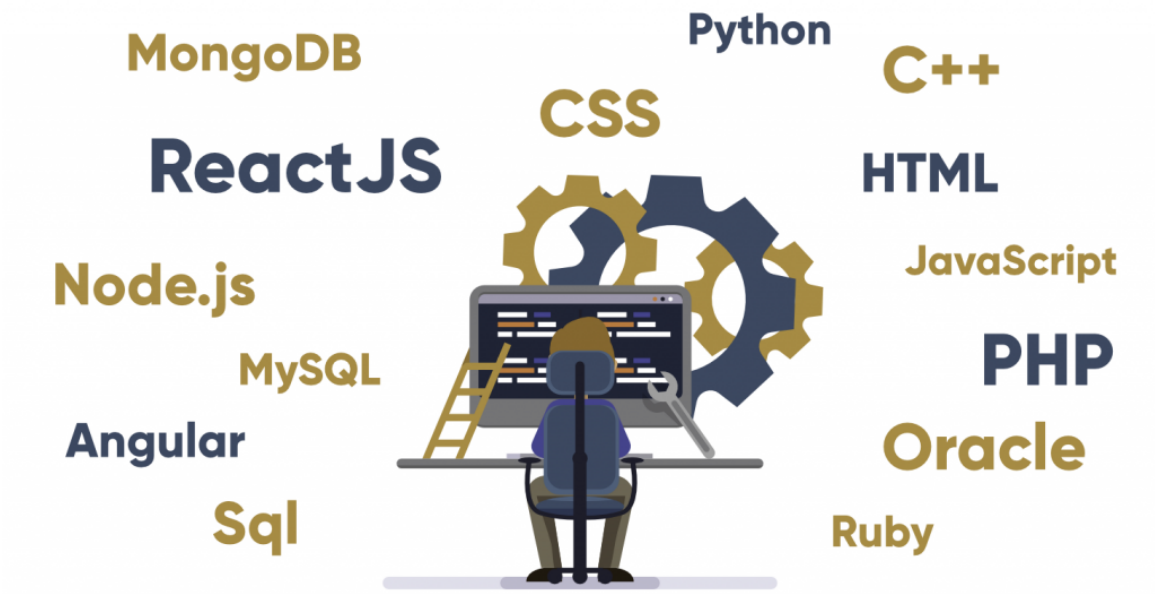
1. https://www.youtube.com/watch?v=_r0VX-aU_T8
2. https://www.youtube.com/watch?v=4nTh3AP6knM
3. https://www.youtube.com/watch?v=j7K7-W82hcE
4. https://www.youtube.com/watch?v=HInixSID4gg
5. https://www.youtube.com/watch?v=9lljOVfOi6o
6. https://www.youtube.com/watch?v=HInixSID4gg&t=26s (with case study example)

W-1 D-6 L-2Hrs

What Is Full-Stack Development?

Full stack development: It refers to the development of both **front end**(client side) and **back end**(server side) portions of any application.

Full stack web Developers: Full stack web developers have the ability to design complete web applications and websites. They work on the frontend, backend, database and debugging of web applications or websites.



Technology related to full stack development:

Front end: It is the visible part of website or web application which is responsible for user experience. The user directly interacts with the front end portion of the web application or website.
Front end Languages: The front end portion is built by using some languages which are discussed below:

- **HTML:**HTML stands for Hyper Text Markup Language. It is used to design the front end portion of web pages using markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. The markup language is used to define the text documentation within tag which defines the structure of web pages.
- **CSS:** Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.
- **JavaScript:**JavaScript is a famous scripting language used to create the magic on the sites to make the site interactive for the user. It is used to enhancing the functionality of a website to running cool games and web-based software.

Front End Frameworks and Libraries:

- **AngularJS:**AngularJs is a JavaScript open source front-end framework that is mainly used to develop single page web applications(SPAs). It is a continuously growing and expanding framework which provides better ways for developing web applications. It changes the static HTML to dynamic HTML. It is an open source project which can be freely used and changed by anyone. It extends HTML attributes with Directives, and data is bound with HTML.
- **React.js:** React is a declarative, efficient, and flexible JavaScript library for building user interfaces. ReactJS is an open-source, component-based front end library responsible only for the view layer of the application. It is maintained by Facebook.
- **Bootstrap:** Bootstrap is a free and open-source tool collection for creating responsive websites and web applications. It is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites.
- jQuery:jQuery is an open source JavaScript library that simplifies the interactions between an HTML/CSS document, or more precisely the Document Object Model (DOM), and JavaScript. Elaborating the terms, jQuery simplifies HTML document traversing and manipulation, browser event handling, DOM animations, Ajax interactions, and cross-browser JavaScript development.
- **SASS:** It is the most reliable, mature and robust CSS extension language. It is used to extend the functionality of an existing CSS of a site including everything from variables, inheritance, and nesting with ease.
- Some other libraries and frameworks are: Semantic-UI, Foundation, Materialize, Backbone.js, Express.js, Ember.js etc.

Other Important Points:

- Work with text editors to use shortcuts and its facilities i.e. Visual studio, Atom, Sublime etc.
- Make UI responsible using grid system.
- Git and git commands like init, add, commit etc for version control and to work with team.
- Other tools like npm& yarn package managers, sass css pre-processor, browser DevTools i.e. chrome devtools.
- Understand using HTTP, JSON, GraphQL APIs to fetch data using axios or other tools.
- It also requires some design skill to make layout and look better.

Back end: It refers to the server-side development of web application or website with a primary focus on how the website works. It is responsible for managing the database through queries and APIs by client-side commands. This type of website mainly consists of three parts front end, back end, and database.

The back end portion is built by using some libraries, frameworks, and languages which are discussed below:

- **PHP: P**HP is a server-side scripting language designed specifically for web development. Since, PHP code executed on server side so it is called server side scripting language.
- **C++** : It is a general purpose programming language and widely used now a days for competitive programming. It is also used as backend language.
- **Java:**Java is one of the most popular and widely used programming language and platform. It is highly scalable. Java components are easily available.
- **Python:** Python is a programming language that lets you work quickly and integrate systems more efficiently.
- **JavaScript**:Javascript can be used as both (front end and back end) programming languages.
- **Node.js:** Node.js is an open source and cross-platform runtime environment for executing JavaScript code outside of a browser. You need to remember that NodeJS is not a framework and it's not a programming language. Most of the people are confused and understand it's a framework or a programming language. We often use Node.js for building back-end services like APIs like Web App or Mobile App. It's used in production by large companies such as Paypal, Uber, Netflix, Walmart and so on.
- **Back End Frameworks:** The list of back end frameworks are: Express, Django, Rails, Laravel, Spring etc.
- The other back end program/scripting languages are: C#, Ruby, REST, GO etc.

Other Important Points:

- Structuring the data in efficient way.
- Handle request-response of APIs for storing and retrieve data.
- Security of data is important.

Note: JavaScript is essential for all stacks as it is dominant technology on Web.

Database: Database is the collection of inter-related data which helps in efficient retrieval, insertion and deletion of data from database and organizes the data in the form of tables, views, schemas, reports etc.

- **Oracle:** Oracle database is the collection of data which is treated as a unit. The purpose of this database is to store and retrieve information related to the query. It is a database server and used to manages information.
- **MongoDB:**MongoDB, the most popular NoSQL database, is an open source document-oriented database. The term 'NoSQL' means 'non-relational'. It means that MongoDB isn't based on the table-like relational database structure but provides an altogether different mechanism for storage and retrieval of data.
- **Sql**: Structured Query Language is a standard Database language which is used to create, maintain and retrieve the relational database.

Popular Stacks:

- **MEAN Stack:**MongoDB, Express, AngularJS and Node.js.
- **MERN Stack:**MongoDB, Express, ReactJS and Node.js
- **Django Stack:**Django, python and MySQL as Database.
- **Rails or Ruby on Rails:** Uses Ruby, PHP and MySQL.
- **LAMP Stack:** Linux, Apache, MySQL and PHP.

What Are The Benefits Of Hiring Full-Stack Developer?

- **They Are All-Rounder:**

Full-stack developers can work on front-end technologies like JavaScript, HTML, CSS, and backend technologies like PHP, Node, Java, Python, and so on. Also, they have knowledge about database and server management, making them all-rounders of web and mobile app development.

- **They Help Reduce Time And Cost:**

Full-stack developers help keep every part of the web application running smoothly. They can help team members solve any issue and greatly reduce infrastructure, personnel, and management cost. As they know numerous technologies/frameworks, they can fix issues quickly and deliver rapidly.

- **They Help Deliver Faster:**

Hiring a full-stack developer reduces the number of persons required to develop an application. Hence, the communication time between team members will also decrease. In bigger teams, they know how to effectively communicate with frontend and backend developers, greatly saving time and cost. This will result in a faster time-to-market of an application.

- **Suitable For Startups And Small Teams:**

Full-stack developers for startups will be proven beneficial as startups require faster prototyping in less time and budget. Collaborative work, innovative thinking, and knowledge of various technologies and approaches make them agile, self-organized, efficient, and excellent assets to the company.

What Are The Disadvantages Of Full-Stack Developer/Development?

- **Jack Of All, Master Of None:**

If there is a requirement to deep dive to solve any issue or need expert advice, full-stack developers might not deliver the best. So many years of experience and constant learning can make them pro over time.

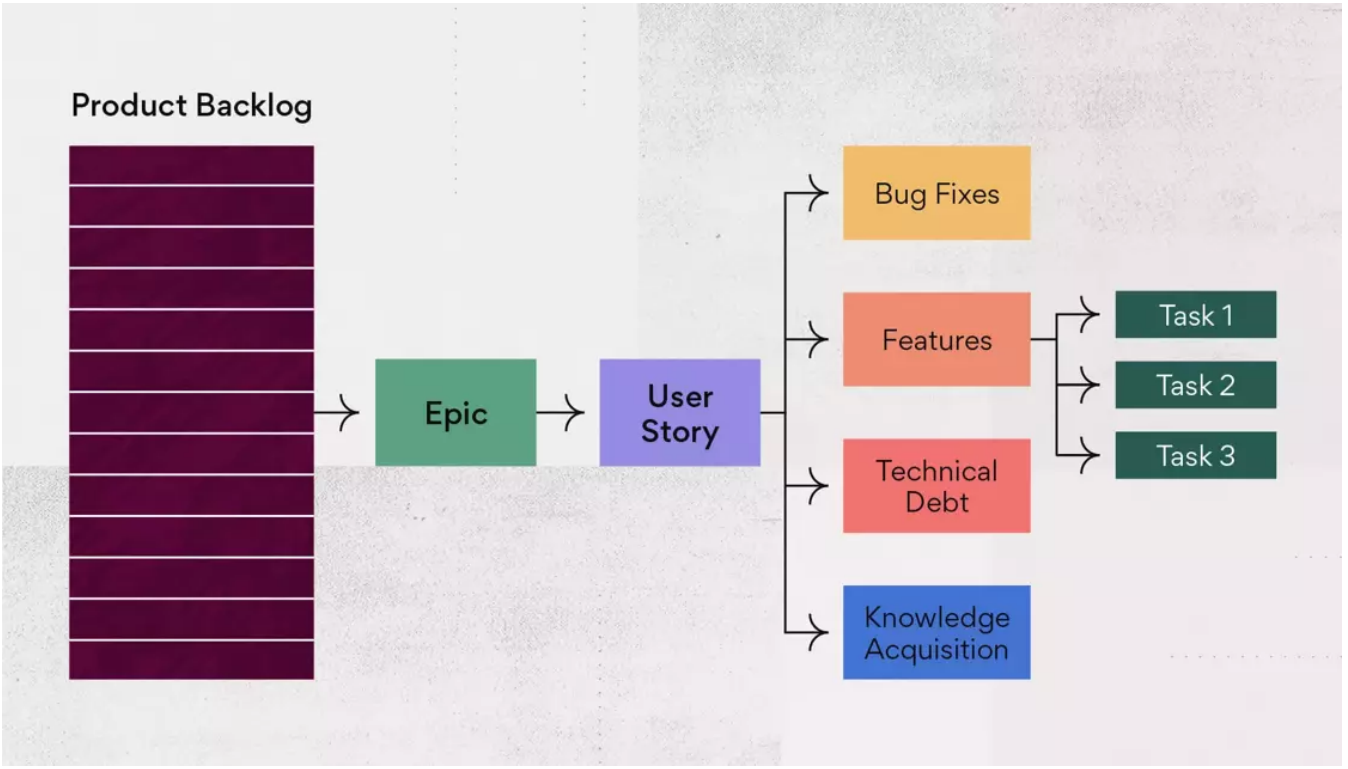
- **Overload Of Tasks:**

It depends on organizational structure and majorly on an individual's capability to complete the tasks. While dependency on the full-stack developer is comparatively higher, sometimes it is difficult to manage and pay equal attention to the number of projects simultaneously.

W-1 D-6 P-3 Hrs

What Is A Product Backlog?

A product backlog is a prioritized list of work items or features that help you meet product goals and set expectations among teams. In general, each product in development should have a dedicated product backlog. Similarly, each product backlog should have a dedicated project team.



.

A product backlog can be created from the product roadmap, which explains the plan of action for the product's evolution. Developers use the tasks in the product backlog to get to their desired outcomes as quickly as possible.

Who uses product backlogs ?

While any developer can use a product backlog, they're most often used by Agile teams. In Agile projects, the teams dedicate their time to product creation and make adjustments as their project progresses. Because of the flexibility of the Agile methodology, tasks on the product backlog aren't set in stone, and you're not expected to complete every one of them. Plus, Agile teams will regularly undergo product backlog refinement to re-prioritize tasks as needed.

What's in a product backlog?

A product backlog commonly includes features, bug fixes, technical debts, and knowledge acquisition. These product backlog items are distinct pieces of work that have yet to be delivered for a product.

1. Features (user stories)
A feature, also known as a user story, is a function of the product that the product user finds valuable. Features can be complex—often referred to as epics—or they can be simple. Creating a story map can help your team determine what the user needs most.

2. Bug fixes
Bug fixes are self-explanatory, and your Scrum team should address these quickly to uphold the integrity of the product. Some bugs may be important enough to interrupt your team's current sprint, while others can wait for the next sprint. An overall rule with bugs, however, is to keep them at the top of your product backlog so your team doesn't forget about them.

3. Technical debts

Technical debt, like financial debt, "accrues interest" when ignored. When developers push technical work to the bottom of the product backlog, it builds up and becomes harder to accomplish. Effective backlog management can prevent the buildup of technical debt. When your team stays organized and takes on technical work in smaller, daily increments, you're less likely to accrue interest on a huge piece of work.

4. Knowledge acquisition

In knowledge acquisition, you gather information to accomplish future tasks. Essentially, this is a research stage. When you identify a feature that needs more research, you create a knowledge acquisition task such as a prototype, experiment, or proof-of-concept to get the information you need to work on the feature.

Steps to create a product backlog

A product backlog is more than a simple to-do list—it's where you break down complex tasks into a series of steps and delegate them to team members. Follow these four steps to develop an effective product backlog.

1. Build a product roadmap

The product roadmap is the foundation for the product backlog. Your team should create a roadmap first, which will then serve as the action plan for how your product will change as it develops. The roadmap is the vision for long-term product development, but it can also evolve.

2. List product backlog items

With your product roadmap in mind, your team can begin listing product backlog items. These items should include both high-priority items and more abstract ideas. During this phase of product backlog creation, you'll also need to communicate with stakeholders and listen to their ideas for product improvements. If you're using the Agile method, you can organize this conversation as part of your sprint planning meeting.

3. Prioritize your backlog

After your team lists all the product backlog items, sort and prioritize your most important tasks. You can identify top-priority items by putting the customer front of mind and considering what items provide the most value to them.

4. Update regularly

As your team works through the product backlog, remember that it's a living document. You can continuously add items to the backlog and prioritize or refine them as you work.

W-2 D-1 L-4Hrs

Software Development (Agile methodology)

- Definegoalofproduct
- Defineepics
- Createroadmapforepics
- Costestimation
- Riskmanagement
- Creatinguserstoriesfortheepic
- CreatingAcceptancecriteria
- sprintplanning
- BacklogRefinement
- SprintDemo

- Burndowncharts
- Sprintretrospective

W-2 D-1 P-3Hrs

- Create and manage product backlog usingappropriatetool likeJira .
- CreateSprint1withrequireduserstories

W-2 D-2 L-2 Hrs P-1 Hrs

Designprinciples

- Availability
- Performance
- Consistency
- Scalability
- Manageability
- costArchitecturalpatterns
- Monolithic
- Layered
- Serviceorientedarchitecture
- Microservicearchitecture

Ref.No4

Step 01 - Need for Architecture - Viewer Page |InfosysSpringboard(onwingspan.com)

W-2 D-3 L-2 Hrs P-2 Hrs

Designmethodsforsecurity

- Applicationsecurity
- Authentication andauthorizationmethodsandtheirusageandconsiderations
- Tokenbased
- Cookiebased
- OpenID
- Thirdpartyaccess
- SAML
- Multifactorauthentication
- Encryption
- Designandimplementauthenticationflowusinganyoneoftheabovelisted.

W-2 D-4 L-1 Hr P-6 Hrs

Designprinciplesfor–UI/UX

CreateUI/UXdesign-forcreateduserstories(wireframing)

Technology,toolsandframeworksfor applicationdevelopment

What is UI design?

User interface (UI) design refers to the **visual design of a product's digital interface**, such as an app screen or website. It's the process of designing the visual and interactive properties of a visual experience, ensuring the interface is both clear for the user to navigate and overall aesthetically pleasing. UI design is a multidisciplinary field involving various elements from visual design, UX design, and graphic design.

Users have become familiar with certain layouts and patterns. From the typography to the color, every aspect of a digital screen influences the user's interaction, and satisfaction, with the app or website. UI design involves the complex art of arranging these elements to guide the user and keep cognitive load to a minimum while creating beautiful, unique experiences that bring the product to life.

UI design is so much more than making things look pretty—especially in today's digitally-led landscape. Most good businesses recognize that exceptional UI design is vital for fostering brand loyalty, recognition, and trust. Customers no longer enjoy well-designed digital experiences; they expect it. In the absence of good design, users will inevitably flee to a competitor.

What is UX design?

UX design, or user experience design, is the process of creating products and experiences that help users get done what they've come to do—as efficiently and effectively as possible.

But what is **user experience**? In short, the user experience is how users feel as they engage with your product. This concept applies to physical objects and daily experiences, like using a coffee maker or buying a car at a dealership; but it also applies to the digital side of life—think about the last time you transferred money using a banking app, purchased something online, or even played a game on your phone.

How each of these apps or websites are designed, combined with your needs and goals as you engage with them, create the user experience. Sometimes it's a really great experience; sometimes, not so much.

UX design is the process of understanding and empathizing with the actual needs and goals of your users, diving into that user experience, locating the problems (often called "pain points"), and finding solutions that result in a better user experience.

For Practical Sessions use the link:
https://careerfoundry.com/en/blog/ux-design/free-wireframing-tools/#wireframecc

Wireframing is a big part of every UX/UI designer's daily job. There is an overwhelming number of wireframe tools out on the market that promise to make creating wireframes quick and easy.

But finding which ones will work best for your designs can be a bit like searching for a needle in a haystack, and finding a wireframing software that fits your price range can add even more stress to your search.

Our top eleven wireframing tools are:

1. Adobe XD
2. Miro
3. Mockplus
4. Balsamiq Wireframes
5. Wireframe.cc
6. Figma
7. Pencil Project
8. NinjaMock
9. FluidUI
10. Mockflow
11. Cacoo

An Introduction to Wireframing with Figma :

https://www.youtube.com/watch?v=FxZFKSgpeBc

(Figma UI Design Tutorial: Get Started in Just 20 Minutes (2022) – Hindi)

https://www.youtube.com/watch?v=hQBXQCepl4M

(Learn Wireframes in Figma in One Video | Create Blog Website Wireframe Tutorial in Hindi 2022)

Wireframing

Wireframes are diagrams that depict the structure of a design, and they can be either low-fidelity (for user research) or mid-fidelity (for UX research).

What are we wireframing?

First, a little background on the UI we'll be building. It will be a table-like structure showing various UX design tools and which step of the UX design workflow each tool is used in. The data will be user-submitted, so the aim is to see which UX *design workflow* is best, rather the overdone "which UI *design tool* is best?"

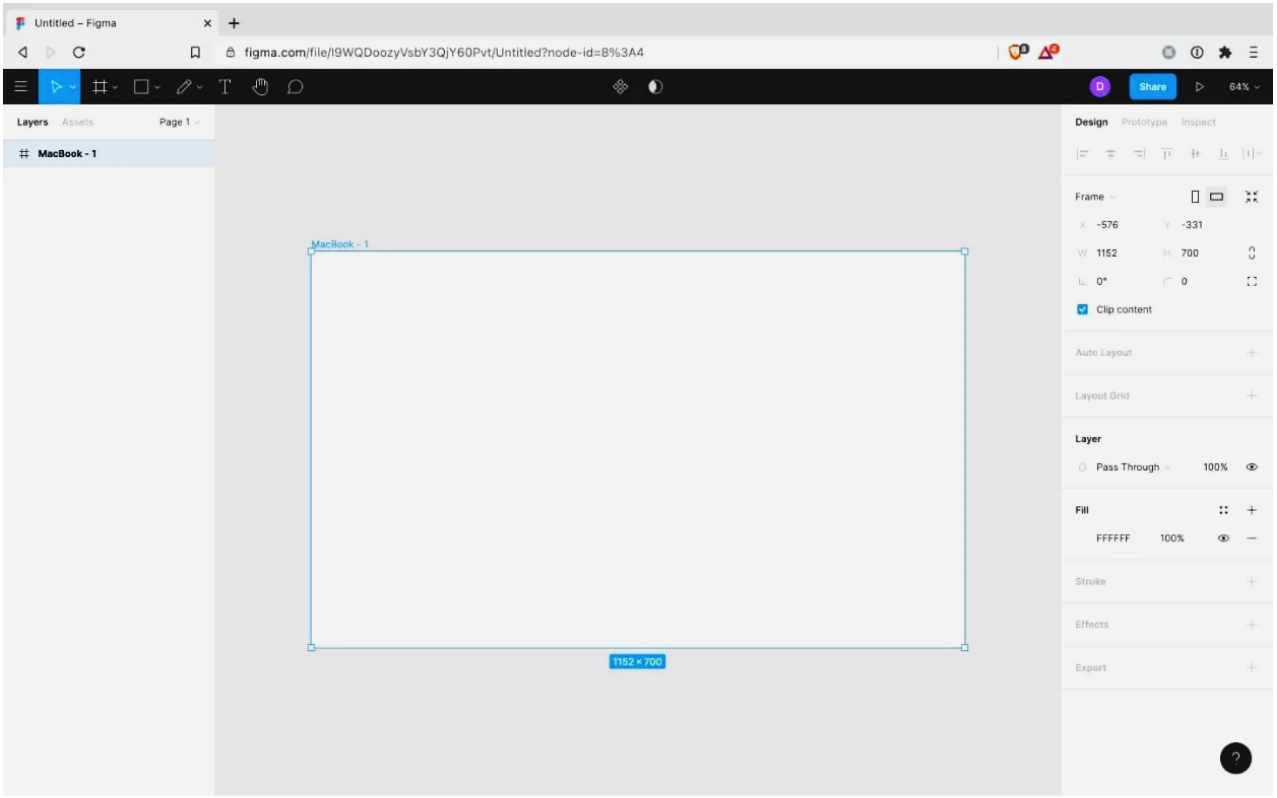
Wireframing will help me to figure out how best to structure this interface without wasting time on figuring out the little visual details. It won't look amazing, but that's fine; it just needs to look nice enough that users can offer me some feedback.

Yes, it's a real UI. At the moment I'm calling it "Toolflows".

Let's begin!

Step 1: Set Up the Artboard

The majority of my website's users are desktop users, so it makes sense to wireframe my design on a desktop artboard. Press `A` on your keyboard, then click *Design>Desktop>MacBook* from the right sidebar of Figma.



Step 2: Gather Functional Requirements

Assuming that you or somebody else did some user research at some stage, we'll need to refer to that to create our wireframe. While conducting user research (*specifically, user interviews, focus groups, and user testing with low-fidelity wireframes*), we would have been made aware of any functional requirements.

Ours are:

- filter by tool
- number of workflow users

Let's start wireframing!

Step 3: Create Text and Shapes

First of all, there *are*Figma wireframe kits available, but I'm not a fan of them exactly. They make me feel constrained to work only with what's available in the kit, so it hinders creativity.

Instead, we'll wireframe using text and shapes.

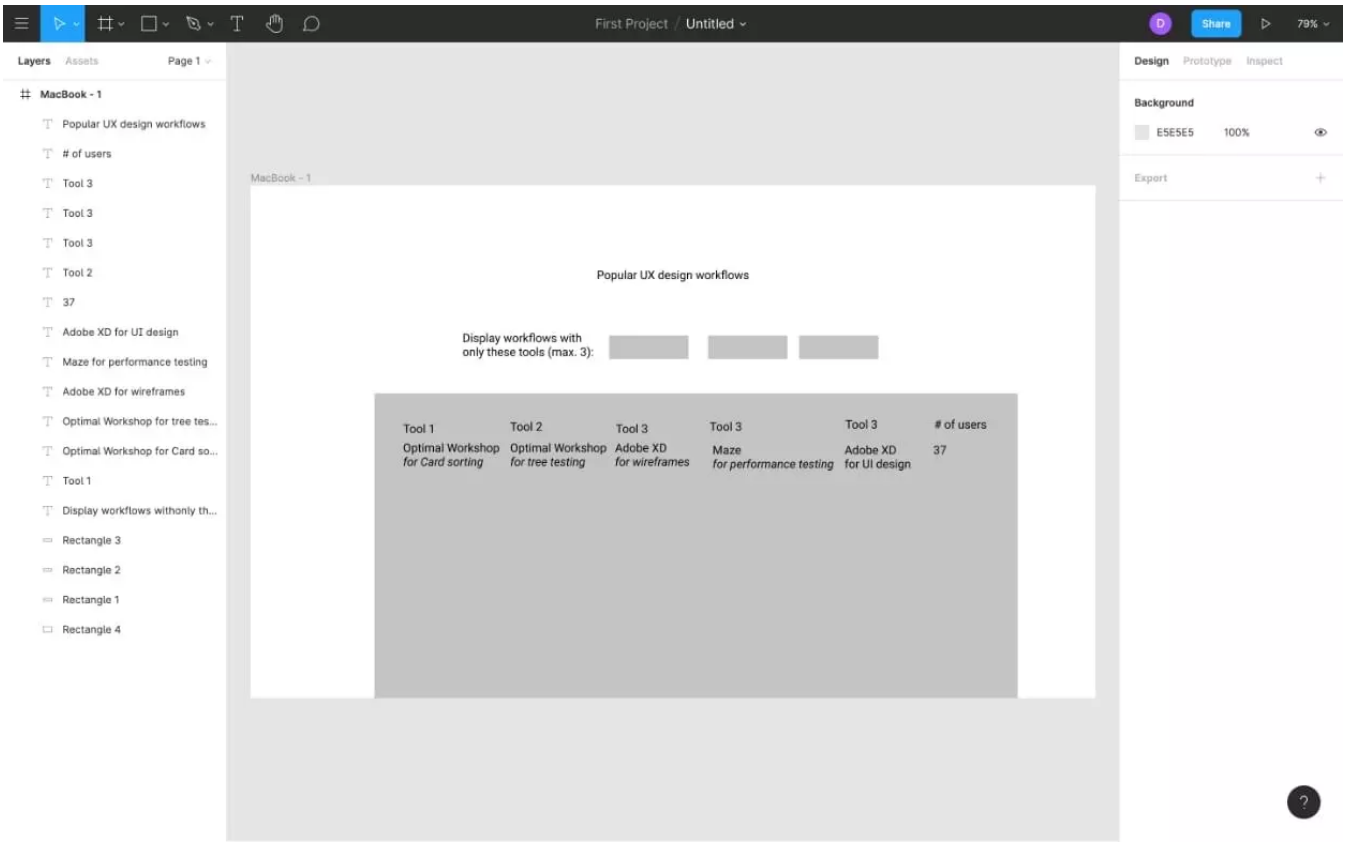
As we learned before with the artboard, the easiest way to create anything in Figma is to abuse the keyboard shortcuts:

- T: Text
- O: Ellipse
- R: Rectangle
- ⇧⌘K: Image
- ⇧L: Arrow
- L: Line

After that, it's simply a case of clicking on the artboard roughly where you'd like the object to appear, and then you can use your mouse and arrow keys to adjust the size and alignment.

Useful shortcuts:

- ⌘-/+ to Zoom.
- ⌘D to Duplicate selected objects.
- ⌘G to Group selected objects (⌘RC to select within).
- Hold ⌘ when mouse-resizing to rotate objects.
- Hold ⇧ when mouse-resizing to maintain aspect ratio.
- Use arrow keys to move objects by 1px (Hold ⇧ for 10px).
- Arrow keys + ⌘ to resize by 1px (Hold ⌘⇧ for 10px).



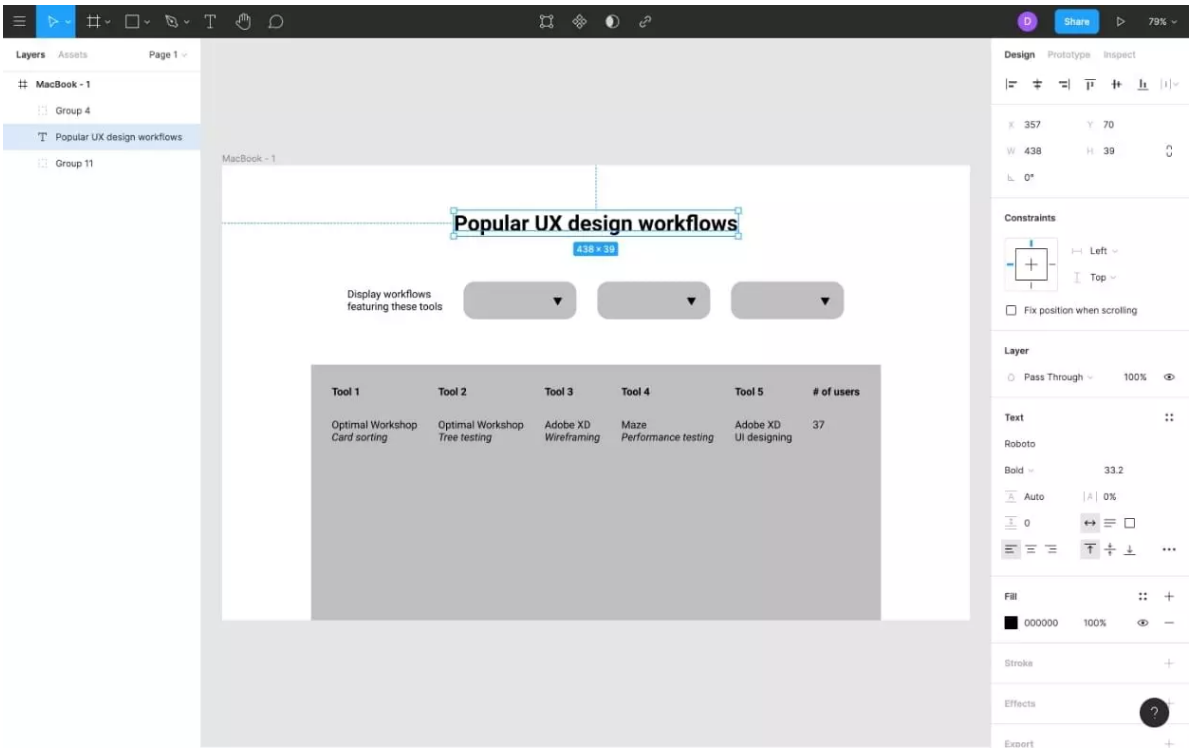
Next, we'll move on to styling.

Step 4: Style, but Don't Style

Using the (hopefully still visible) *Design* sidebar, we can alter the styles of objects on our artboard, both aesthetically and to specify the sizing and alignment of them more accurately.

Whether you're using the *Design* sidebar or the arrow keys to size/align, hold ⌥ (option) to measure the distance between objects.

Remember, don't design much beyond sizing and aligning. Give rounded corners to buttons (so that we can very clearly see that they're buttons), bolden headings, etc. Clarity, not aesthetics.

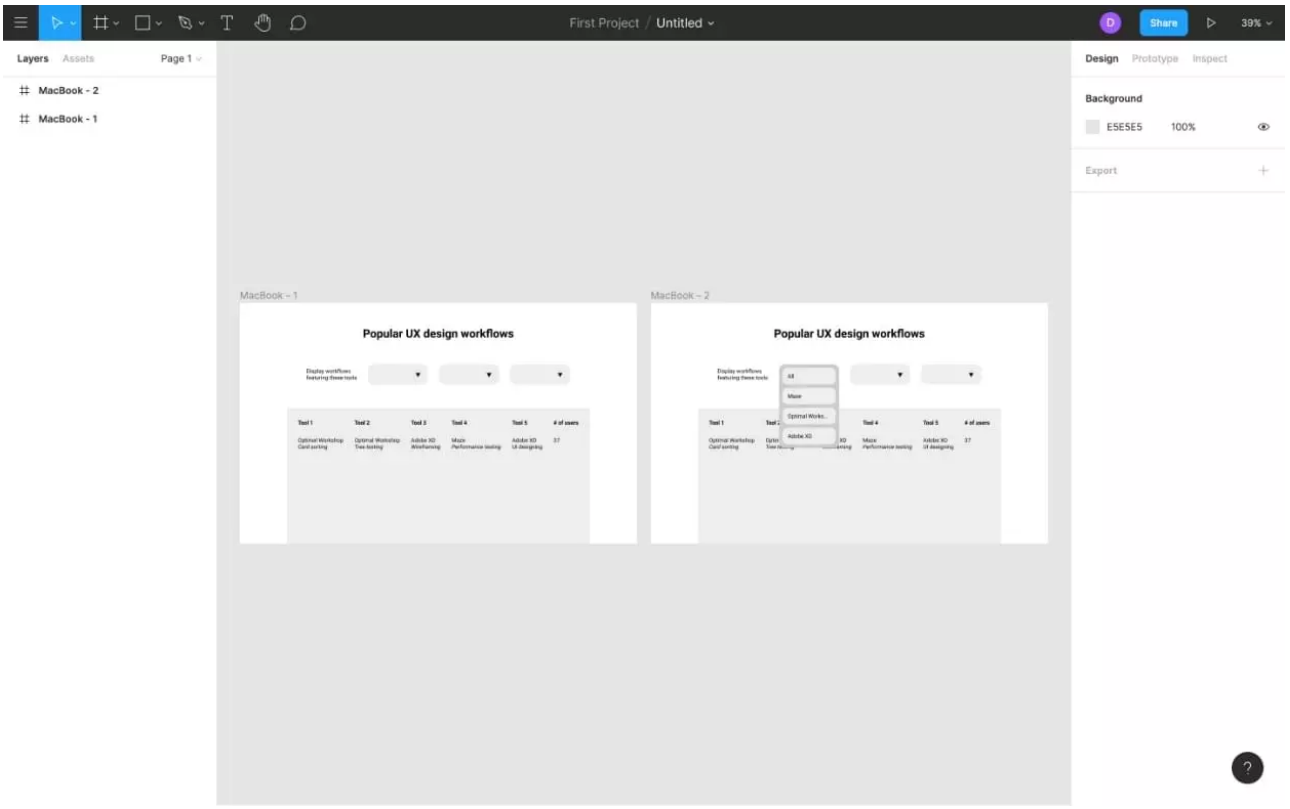


And that's it; now we have a wireframe. That being said, for designs that require interaction, we'll want to demonstrate how exactly our design would function, so without further ado let's move onto *prototyping*. Prototyping is the step where we make the design interactive (that is, feel as if it were the real thing).

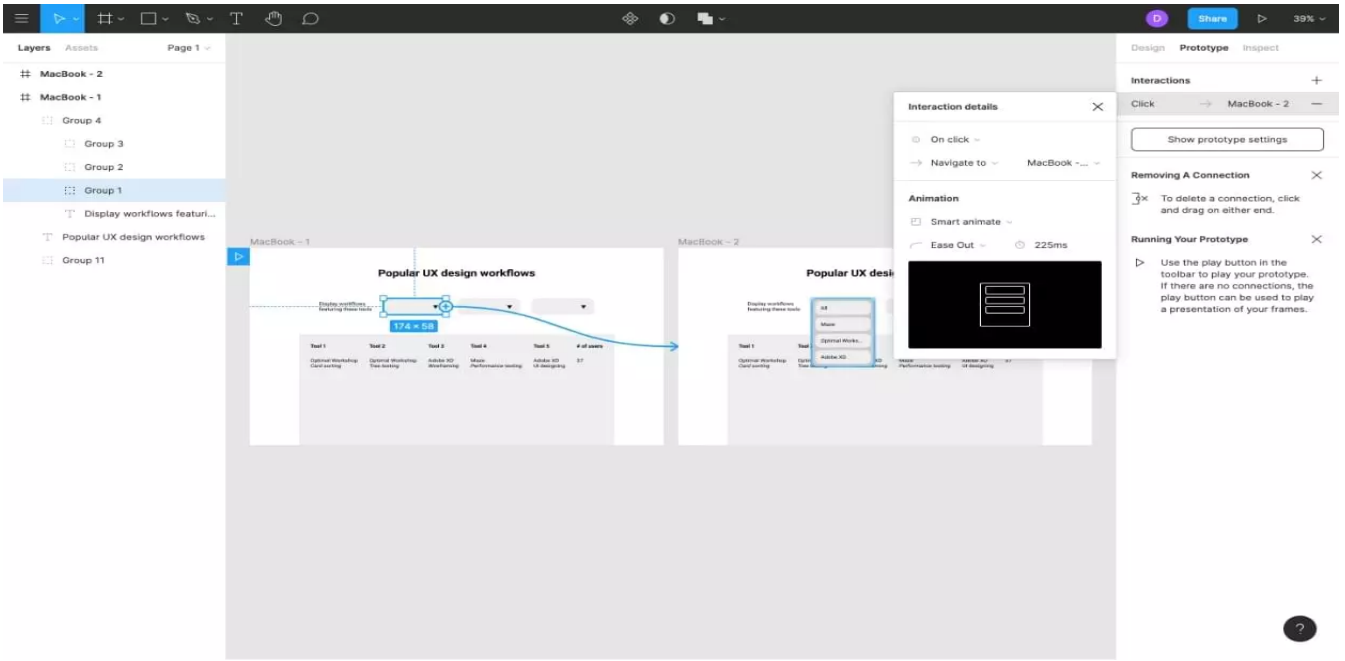
Step 6: Create Transitions

The concept behind this step is to duplicate our artboard, demonstrate how we want our design to look at the end of the interaction, and then define the trigger (animation optional) that will initiate the transition between the two "states".

Start by duplicating the artboard (⌘D), and then change whatever needs to be changed in this new artboard. In my case, I want to show the dropdown menu, which filters workflows by tool.



Next, switch to the *Prototype* sidebar before selecting the object that will be the trigger for your interaction. For me, this is the *closed* dropdown menu from the original artboard.
Once selected, there should be a draggable circular + icon. Drag this circle onto the second artboard to create a “connection”.

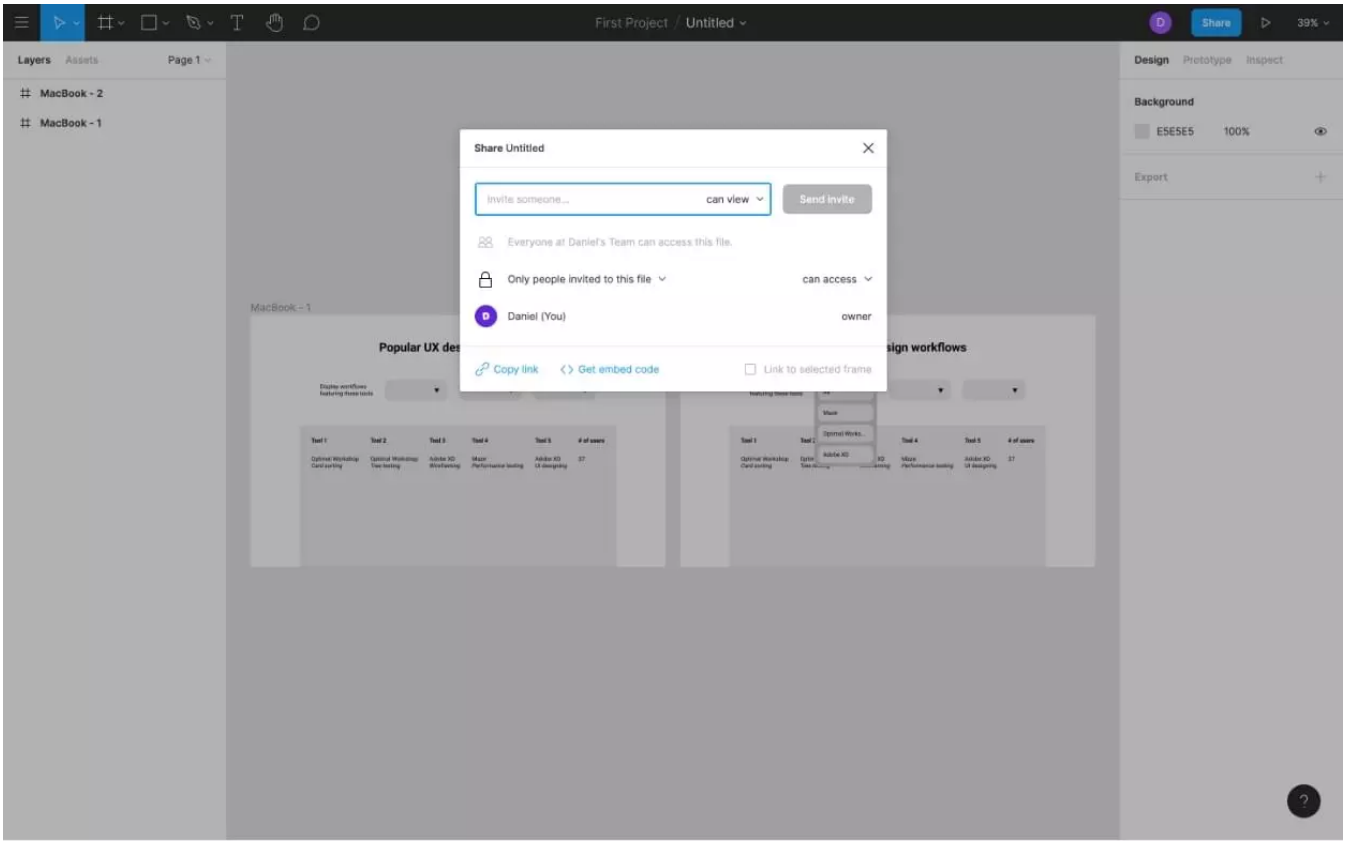


Step 7: Set Animation (Optional)
Next, we should set some animation for the interaction so that users can more easily see what's changed between the two states.
Animations don't need to be fancy, since we're only designing with mid-fidelity right now, so from the *Interaction details* dialog that revealed itself after creating the connection, set the *Animation* to *Smart animate*. Smart animate animates layers that didn't exist in the "trigger artboard". Smart, ay?

If you're prototyping with mid-fidelity wireframes, you won't really need to tinker with any of the other options, but it's cool that you know how to create animated connections now.

Step 8: Test, Test, Test
Next, we'll want to test it, first to ensure that all of the connections work, then to acquire what I'd consider "low-level feedback" from stakeholders. Obviously the real value comes from testing with users, but that doesn't mean that our stakeholders don't have anything to contribute. Maybe we missed something?

To share with stakeholders, hit the *Share* button in the top-right corner. Stakeholders will be able to leave some comments.



For your own testing (which I'd recommend doing before sending sharing with anybody else) you'll want to download Figma Mirror for iOSor Android. When designing for desktop, though, simply hit the *play* icon in the top-right to enter "Present" mode.

Bonus Step 9: Conduct Usability Tests

If you'd like to acquire more qualitative feedback on your design (task completion rate, time to completion, etc.), then apps like Maze and Useberry (which both work with Figma) will help you to accomplish exactly that. After all, this is why we're wireframing, right? To make our design more usable.

Conclusion: Figma Has It All

Figma has it all, including a thriving and dedicated community made up of both macOS and Windows designers, and even those that simply want to design in their web browser (so, Linux too!).

So, what now? Well, you could explore the Figma community, see what they're making, and maybe even download some Figma plugins to extend and automate your UI design workflow.

W-3 D-2 L-2 Hrs P-5 Hrs

Configuration management or version control system:

What is a "version control system"?
Version control systems are a category of software tools that helps in recording changes made to files by keeping a track of modifications done in the code.

Why Version Control system is so Important?
As we know that a software product is developed in collaboration by a group of developers they might be located at different locations and each one of them contributes to some specific kind of functionality/features. So in order to contribute to the product, they made modifications to the source code(either by adding or removing). A version control system is a kind of software that helps the developer team to efficiently communicate and manage(track) all the changes that have been made to the source code along with the information like who made and what changes have been made.

Basically Version control system keeps track on changes made on a particular software and take a snapshot of every modification. Let's suppose if a team of developer add some new functionalities in an application and the updated version is not working properly so as the version control system keeps track of our work so with the help of version control system we can omit the new changes and continue with the previous version.

Benefits of the version control system:
- Enhances the project development speed by providing efficient collaboration,
- Leverages the productivity, expedites product delivery, and skills of the employees through better communication and assistance,
- Reduce possibilities of errors and conflicts meanwhile project development through traceability to every small change,
- Employees or contributors of the project can contribute from anywhere irrespective of the different geographical locations through this VCS,
- For each different contributor to the project, a different working copy is maintained and not merged to the main file unless the working copy is validated. The most popular example is Git, Helix core, Microsoft TFS,

- Helps in recovery in case of any disaster or contingent situation,
- Informs us about Who, What, When, Why changes have been made.

Use of Version Control System:

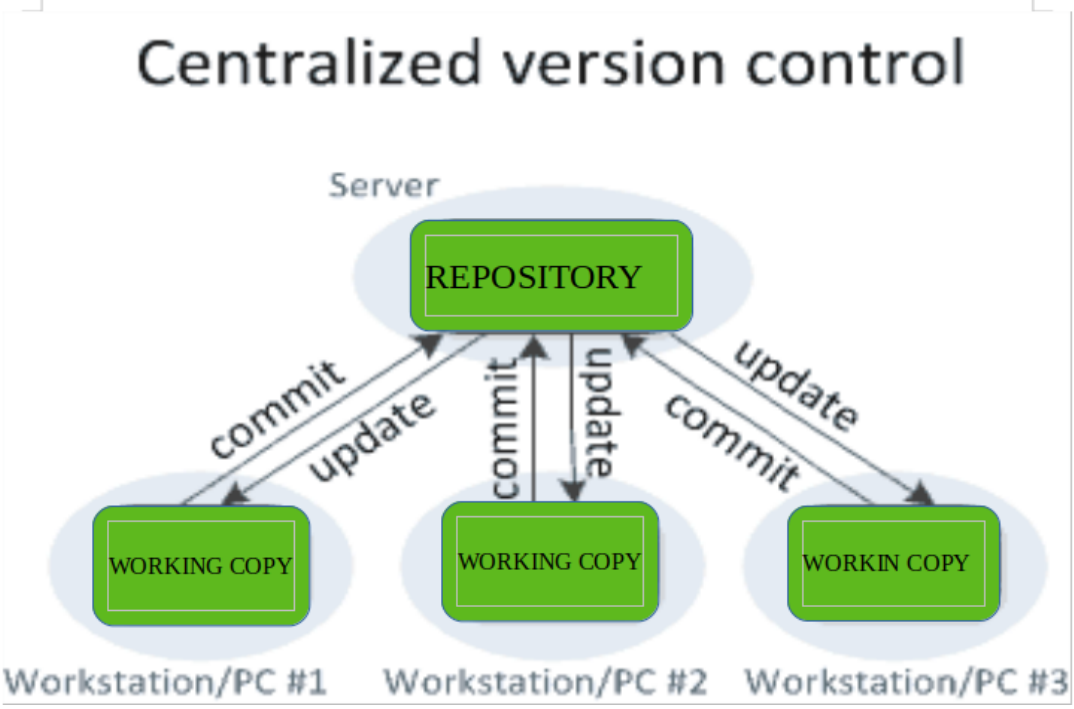
- **A repository**: It can be thought of as a database of changes. It contains all the edits and historical versions (snapshots) of the project.
- **Copy of Work** (sometimes called as checkout): It is the personal copy of all the files in a project. You can edit to this copy, without affecting the work of others and you can finally commit your changes to a repository when you are done making your changes.
- **Working in a group:** Version control helps you with the, merging different requests to main repository without making any undesirable changes. You may test the functionalities without putting it live, and you don't need to download and set up each time, just pull the changes and do the changes, test it and merge it back.

Types of Version Control Systems:

- Local Version Control Systems
- Centralized Version Control Systems
- Distributed Version Control Systems

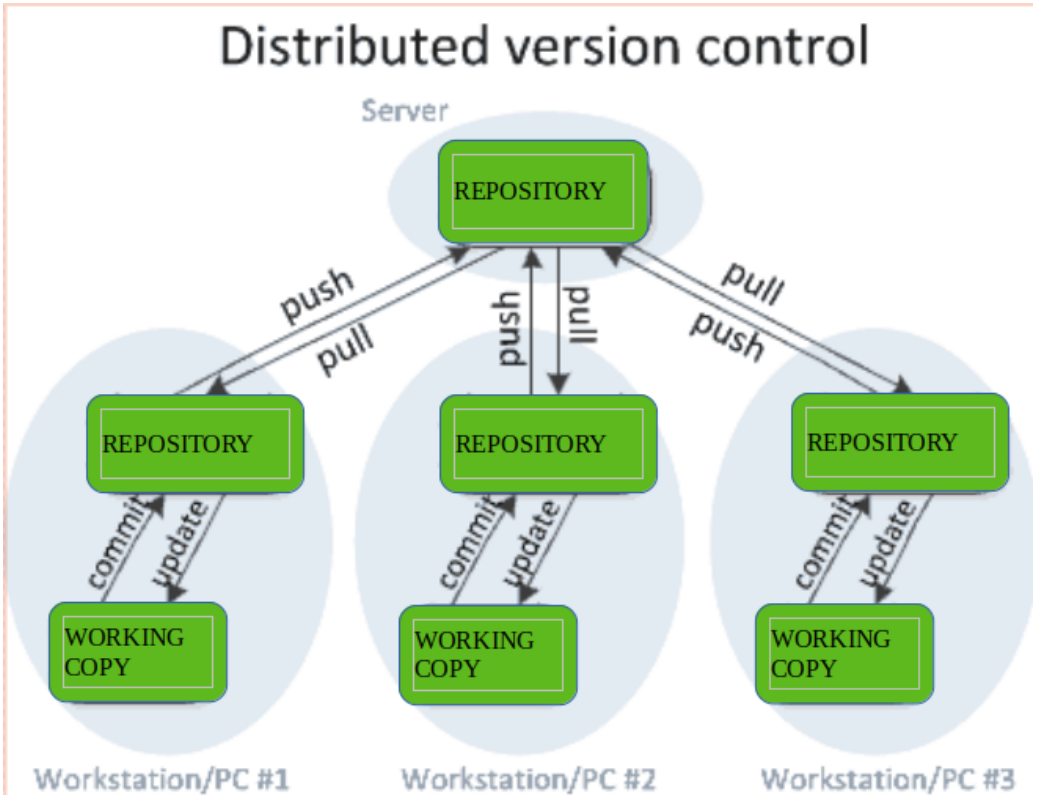
Local Version Control Systems: It is one of the simplest forms and has a database that kept all the changes to files under revision control. RCS is one of the most common VCS tools. It keeps patch sets (differences between files) in a special format on disk. By adding up all the patches it can then re-create what any file looked like at any point in time.

Centralized Version Control Systems: Centralized version control systems contain just one repository globally and every user need to commit for reflecting one's changes in the repository. It is possible for others to see your changes by updating.



Distributed Version Control Systems: Distributed version control systems contain multiple repositories. Each user has their own repository and working copy. Just committing your changes will not give others access to your changes. This is because commit will reflect those changes in your local repository and you need to push them in order to make them visible on the central repository. Similarly,

When you update, you do not get others' changes unless you have first pulled those changes into your repository.
The most popular distributed version control systems are Git, and Mercurial. They help us overcome the problem of single point of failure.



Purpose of Version Control:

- Multiple people can work simultaneously on a single project. Everyone works on and edits their own copy of the files and it is up to them when they wish to share the changes made by them with the rest of the team.
- It also enables one person to use multiple computers to work on a project, so it is valuable even if you are working by yourself.
- It integrates the work that is done simultaneously by different members of the team
- Version control provides access to the historical versions of a project.

FundamentalsofGit

Git is a free and open-source version control system, originally created by Linus Torvalds in 2005. Unlike older centralized version control systems such as SVN and CVS, Git is distributed: every developer has the full history of their code repository locally. This makes the initial clone of the repository slower, but subsequent operations such as commit, blame, diff, merge, and log dramatically faster.

Git also has excellent support for branching, merging, and rewriting repository history, which has led to many innovative and powerful workflows and tools. Pull requests are one such popular tool that allows teams to collaborate on Git branches and efficiently review each other's code. Git is the most widely

used version control system in the world today and is considered the modern standard for software development.

How Git works
Here is a basic overview of how Git works:

1. Create a "repository" (project) with a git hosting tool (like Bitbucket)
2. Copy (or clone) the repository to your local machine
3. Add a file to your local repo and "commit" (save) the changes
4. "Push" your changes to your main branch
5. Make a change to your file with a git hosting tool and commit
6. "Pull" the changes to your local machine
7. Create a "branch" (version), make a change, commit the change
8. Open a "pull request" (propose changes to the main branch)
9. "Merge" your branch to the main branch

GitClientinstallationandsetup

Install Git on Windows

Git for Windows stand-alone installer

1. Download the latest Git for Windows installer. (https://git-scm.com/download/win)
2. When you've successfully started the installer, you should see the **Git Setup** wizard screen. Follow the **Next** and **Finish** prompts to complete the installation. The default options are pretty sensible for most users.
3. Open a Command Prompt (or Git Bash if during installation you elected not to use Git from the Windows Command Prompt).
4. Run the following commands to configure your Git username and email using the following commands, replacing Emma's name with your own. These details will be associated with any commits that you create:

```
$ gitconfig --global user.name"Ajay Angadi"
$ gitconfig --globaluser.email"akangadi09@gmail.com"
```

5. *Optional: Install the Git credential helper on Windows*
   Bitbucket supports pushing and pulling over HTTP to your remote Git repositories on Bitbucket. Every time you interact with the remote repository, you must supply a username/password combination. You can store these credentials, instead of supplying the combination every time, with the Git Credential Manager for Windows.

Learn Git with Bitbucket Cloud

Create a Git repository

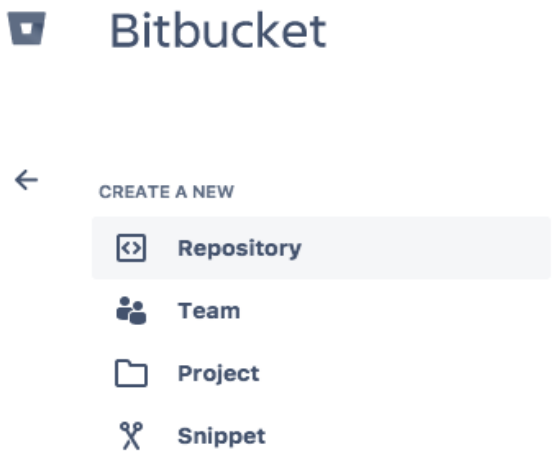
As our new Bitbucket space station administrator, you need to be organized. When you make files for your space station, you'll want to keep them in one place and shareable with teammates, no matter where they are in the universe. With Bitbucket, that means adding everything to a repository. Let's create one!

- ***Some fun facts about repositories***
    - You have access to all files in your local repository, whether you are working on one file or multiple files.
    - You can view public repositories without a Bitbucket account if you have the URL for that repository.
    - Each repository belongs to a user account or a team. In the case of a user account, that user owns the repository. + In the case of a team, that team owns it.
    - The repository owner is the only person who can delete the repository. If the repository belongs to a team, an admin can delete the repository.
    - A code project can consist of multiple repositories across multiple accounts but can also be a single repository from a single account.
    - Each repository has a 2 GB size limit, but we recommend keeping your repository no larger than 1 GB.

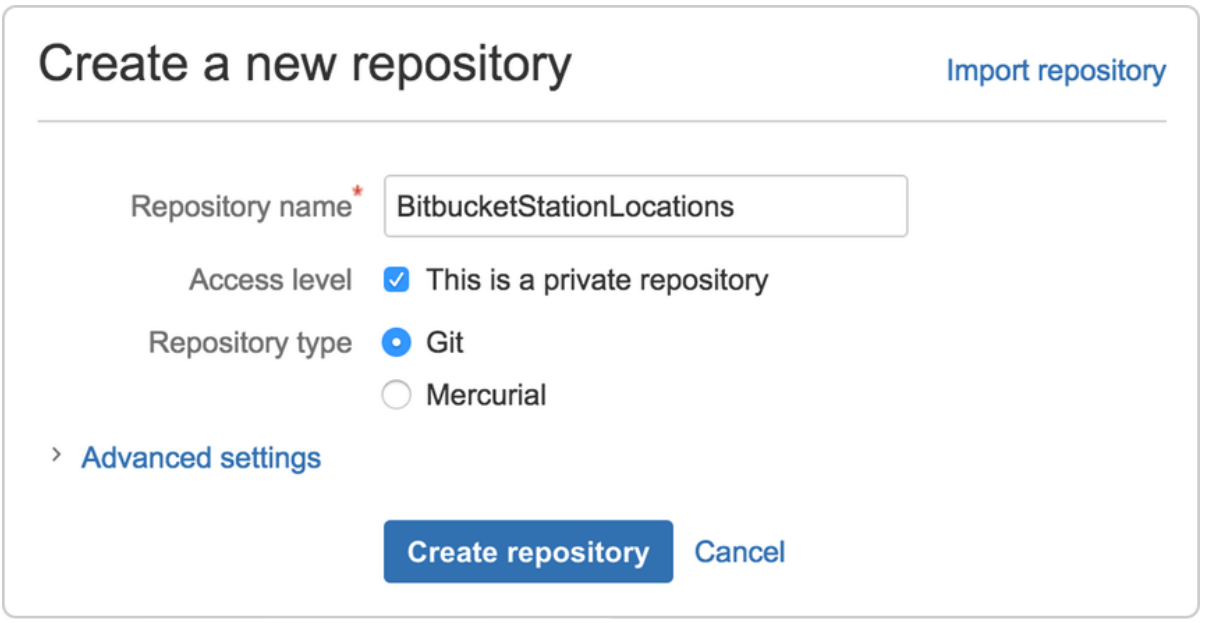
Step 1. Create the repository : Initially, the repository you create in Bitbucket (https://bitbucket.org/) is going to be empty without any code in it. That's okay because you will start adding some files to it soon. This Bitbucket repository will be the central repository for your files, which means that others can access that repository if you give them permission. After creating a repository, you'll copy a version to your local system—that way you can update it from one repo, then transfer those changes to the other.

Do the following to create your repository:

1. From Bitbucket, (https://bitbucket.org/)click the + icon in the global sidebar and select **Repository**.



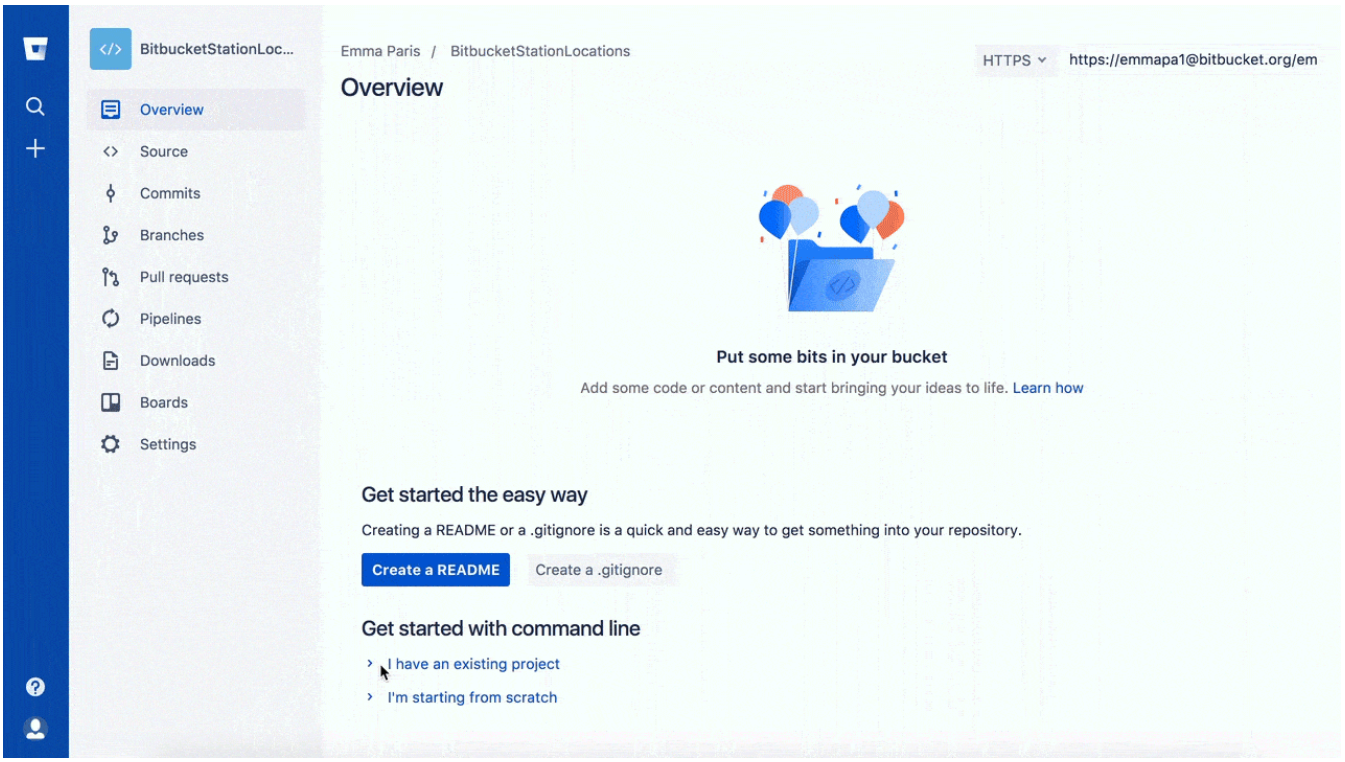
Bitbucket displays the **Create a new repository** page. Take some time to review the dialog's contents. With the exception of the **Repository type**, everything you enter on this page you can later change.



2. Enter `BitbucketStationLocations` for the **Name** field. Bitbucket uses this **Name** in the URL of the repository. For example, if the user `the_best` has a repository called `awesome_repo`, the URL for that repository would be `https://bitbucket.org/the_best/awesome_repo`.
3. For **Access level**, leave the **This is a private repository box** checked. A private repository is only visible to you and those you give access to. If this box is unchecked, everyone can see your repository.
4. Pick **Git** for the **Repository type**. Keep in mind that you can't change the repository type after you click **Create repository**.
5. Click **Create repository**. Bitbucket creates your repository and displays its **Overview** page.

Step 2. Explore your new repository

Take some time to explore the repository you have just created. You should be on the repository's **Overview** page:



Click + from the global sidebar for common actions for a repository. Click items in the navigation sidebar to see what's behind each one, including Settings to update repository details and other settings. To view the shortcuts available to navigate these items, press the **?**key on your keyboard.

When you click the **Commits** option in the sidebar, you find that you have no commits because you have not created any content for your repository. Your repository is private and you have not invited anyone to the repository, so the only person who can create or edit the repository's content right now is you, the repository owner.

Copy your Git repository and add files

Now that you have a place to add and share your space station files, you need a way to get to it from your local system. To set that up, you want to copy the Bitbucket repository to your system. Git refers to copying a repository as "cloning" it. When you clone a repository, you create a connection between the Bitbucket server (which Git knows as origin) and your local system.

Step 1. Clone your repository to your local system

Open a browser and a terminal window from your desktop. After opening the terminal window, do the following:

1. Navigate to your home (~) directory

```
$ cd ~
```

1. As you use Bitbucket more, you will probably work in multiple repositories. For that reason, it's a good idea to create a directory to contain all those repositories.

2. Create a directory to contain your repositories.

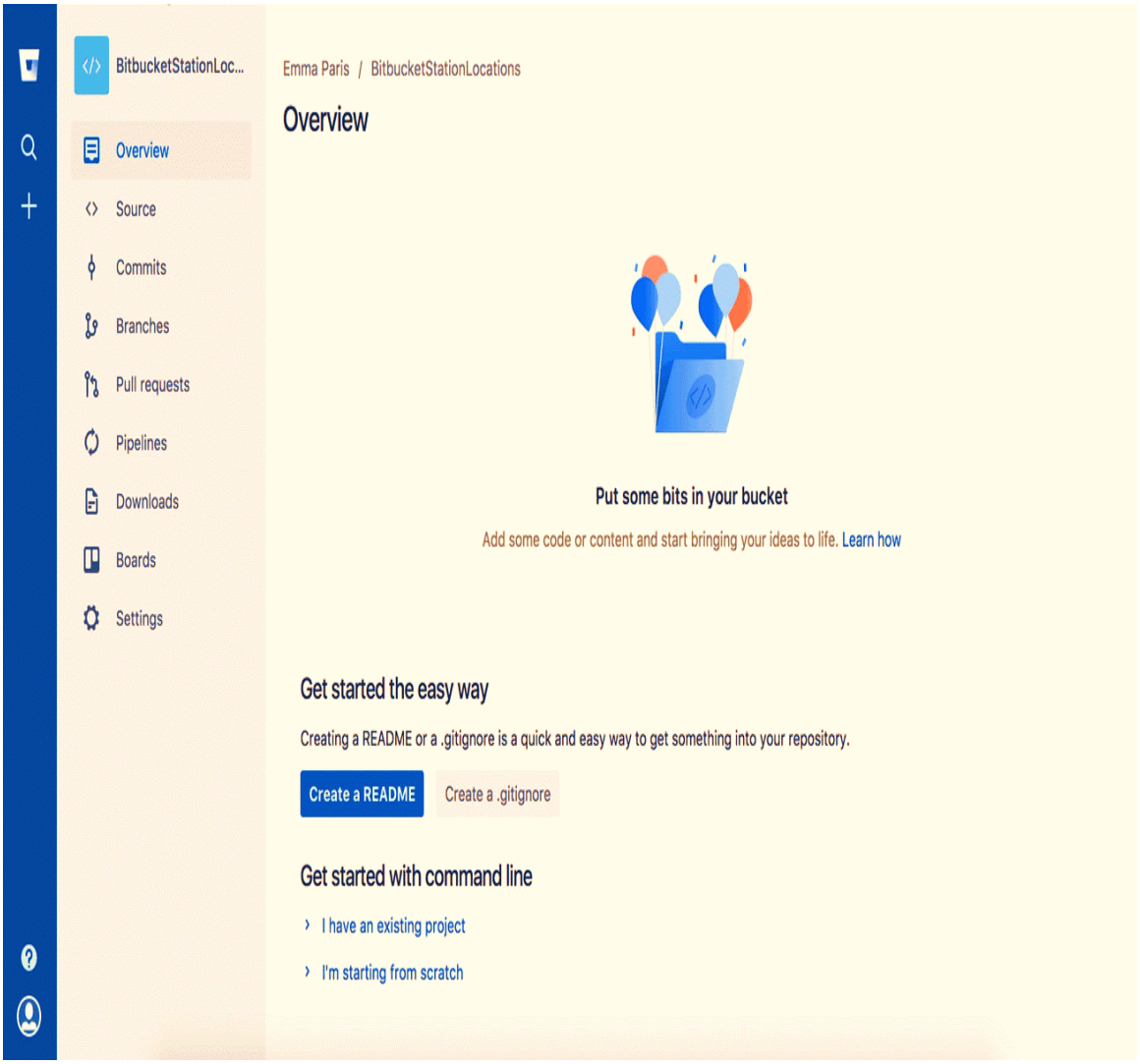
```
$ mkdir repos
```

3. From the terminal, update the directory you want to work in to your new repos directory.

```
$ cd ~/repos
```

4. From Bitbucket, go to your **BitbucketStationLocations** repository.
5. Click the + icon in the global sidebar and select **Clone this repository**.

   Bitbucket displays a pop-up clone dialog. By default, the clone dialog sets the protocol to **HTTPS** or **SSH**, depending on your settings. For the purposes of this tutorial, don't change your default protocol.



6. Copy the highlighted clone command.
7. From your terminal window, paste the command you copied from Bitbucket and press **Return**.
8. Enter your Bitbucket password when the terminal asks for it. If you created an account by linking to Google, use your password for that account.
   - If you experience a **Windows password error**:

- In some versions of Microsoft Windows operating system and Git you might see an error similar to the one in the following example.

Windows clone password error example

```
$ git clonehttps://emmap1@bitbucket.org/emmap1/bitbucketstationlocations.git

       Cloning into 'bitbucketspacestation'...

       fatal: could not read

       Password for 'https://emmap1@bitbucket.org': No such file or directory
```

- If you get this error, enter the following at the command line:

```
 $ git config --global core.askpass
```

- Then go back to step 4 and repeat the clone process. The bash agent should now prompt you for your password. You should only have to do this once.

o At this point, your terminal window should look similar to this:

```
$ cd ~/repos

$ git clone https://emmap1@bitbucket.org/emmap1/bitbucketstationlocations.git
Cloning into 'bitbucketstationlocations'...
Password
warning: You appear to have cloned an empty repository.
```

o You already knew that your repository was empty right? Remember that you have added no source files to it yet.

9. List the contents of your repos directory and you should see your `bitbucketstationlocations` directory in it.

```
$ ls
```

Congratulations! You've cloned your repository to your local system.

Step 2. Add a file to your local repository and put it on Bitbucket

With the repository on your local system, it's time to get to work. You want to start keeping track of all your space station locations. To do so, let's create a file about all your locations.

1. Go to your terminal window and navigate to the top level of your local repository.

```
$ cd ~/repos/bitbucketstationlocations/
```

2. Enter the following line into your terminal window to create a new file with content.

```
$ echo "Earth's Moon" >> locations.txt
```

If the command line doesn't return anything, it means you created the file correctly!

3. Get the status of your local repository. The `git status` command tells you about how your project is progressing in comparison to your Bitbucket repository.

At this point, Git is aware that you created a new file, and you'll see something like this:

```
$ git status
 On branch main
 Initial commit
 Untracked files:
   (use "git add <file>..." to include in what will be committed)
     locations.txt
 nothing added to commit but untracked files present (use "git add" to track)
```

The file is untracked, meaning that Git sees a file not part of a previous commit. The status output also shows you the next step: adding the file.

4. Tell Git to track your new `locations.txt` file using the `git add` command. Just like when you created a file, the `git add` command doesn't return anything when you enter it correctly.

```
$ git add locations.txt
```

The `git add` command moves changes from the working directory to the Git staging area. The staging area is where you prepare a snapshot of a set of changes before committing them to the official history.

5. Check the status of the file.

```
$ git status
On branch main
Initial commit
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file: locations.txt
```

Now you can see the new file has been added (staged) and you can commit it when you are ready. The `git status` command displays the state of the working directory and the staged snapshot.

6. Issue the `git commit` command with a commit message, as shown on the next line. The -m indicates that a commit message follows.

```
$ git commit -m 'Initial commit'
 [main (root-commit) fedc3d3] Initial commit
  1 file changed, 1 insertion(+)
  create mode 100644 locations.txt
```

The `git commit` takes the staged snapshot and commits it to the project history. Combined with `git add`, this process defines the basic workflow for all Git users.

Up until this point, everything you have done is on your local system and invisible to your Bitbucket repository until you push those changes.

- *Learn a bit more about Git and remote repositories*

- Git's ability to communicate with remote repositories (in your case, Bitbucket is the remote repository) is the foundation of every Git-based collaboration workflow.
- Git's collaboration model gives every developer their own copy of the repository, complete with its own local history and branch structure. Users typically need to share a series of commits rather than a single changeset. Instead of committing a changeset from a working copy to the central repository, Git lets you share entire branches between repositories.

- You manage connections with other repositories and publish local history by "pushing" branches to other repositories. You see what others have contributed by "pulling" branches into your local repository.

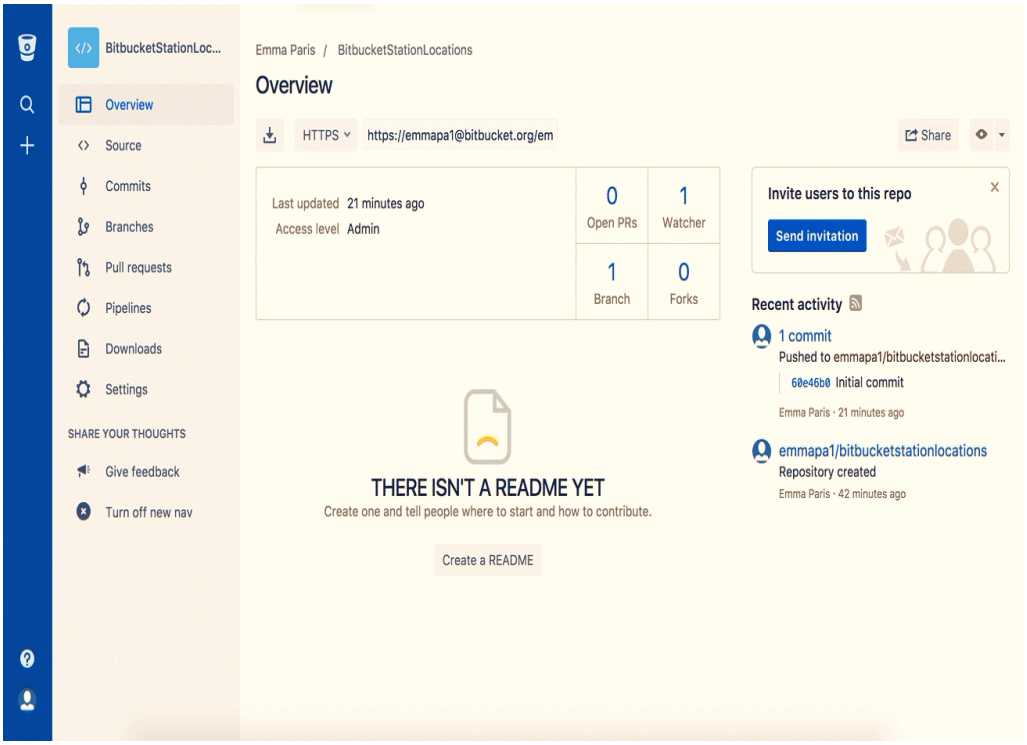
7. Go back to your local terminal window and send your committed changes to Bitbucket using `git push origin main`. This command specifies that you are pushing to the main branch (the branch on Bitbucket) on origin (the Bitbucket server).

   You should see something similar to the following response:

```
$ git push origin main
Counting objects: 3, done.
Writing objects: 100% (3/3), 253 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0) To https://emmap1@bitbucket.org/emmap1/
bitbucketstationlocations.git
  * [new branch] main -> main
Branch main set up to track remote branch main from origin.
```

Your commits are now on the remote repository (origin).

8. Go to your **BitbucketStationLocations** repository on Bitbucket.
9. If you click **Commits** in the sidebar, you'll see a single commit on your repository. Bitbucket combines all the things you just did into that commit and shows it to you. You can see that the Author column shows the value you used when you configured the Git global file `( ~/.gitconfig)`.
   If you click **Source** in the sidebar, you'll see that you have a single source file in your repository, the `locations.txt` file you just added.



Remember how the repository looked when you first created it? It probably looks a bit different now.

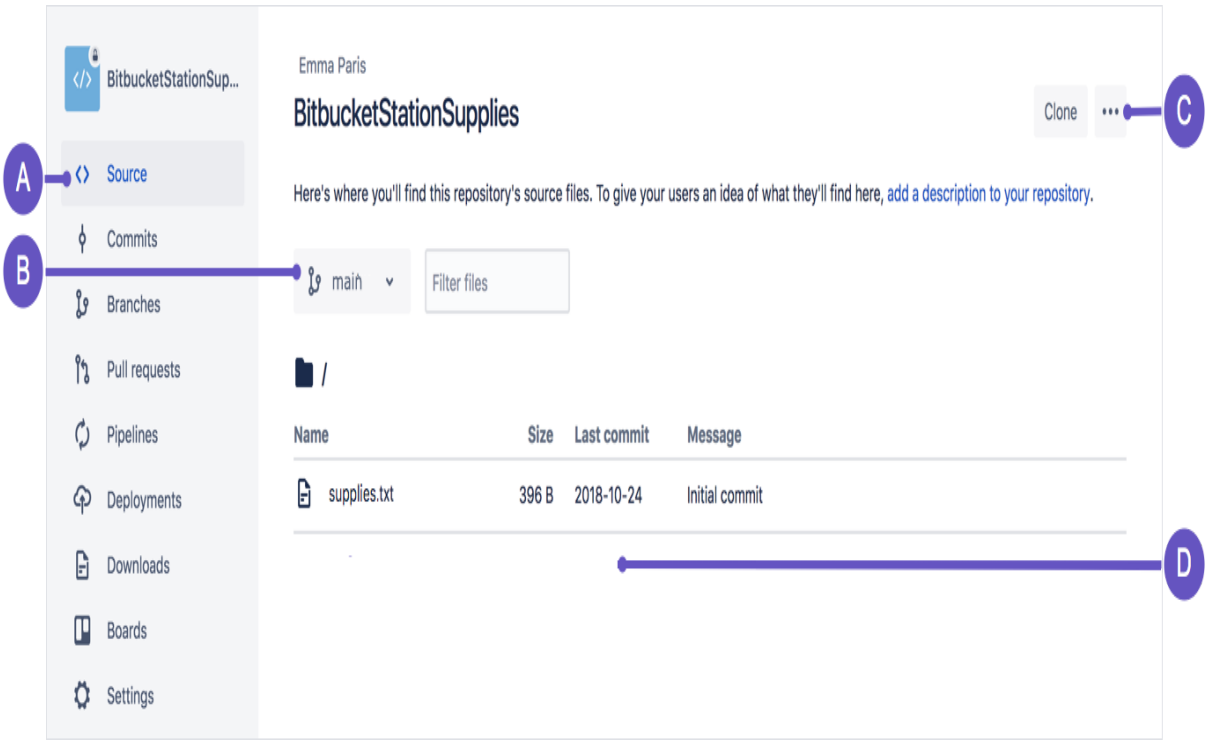
Pull changes from your Git repository on Bitbucket Cloud

Next on your list of space station administrator activities, you need a file with more details about your locations. Since you don't have many locations at the moment, you are going to add them right from Bitbucket.

Step 1. Create a file in Bitbucket

To add your new locations file, do the following:

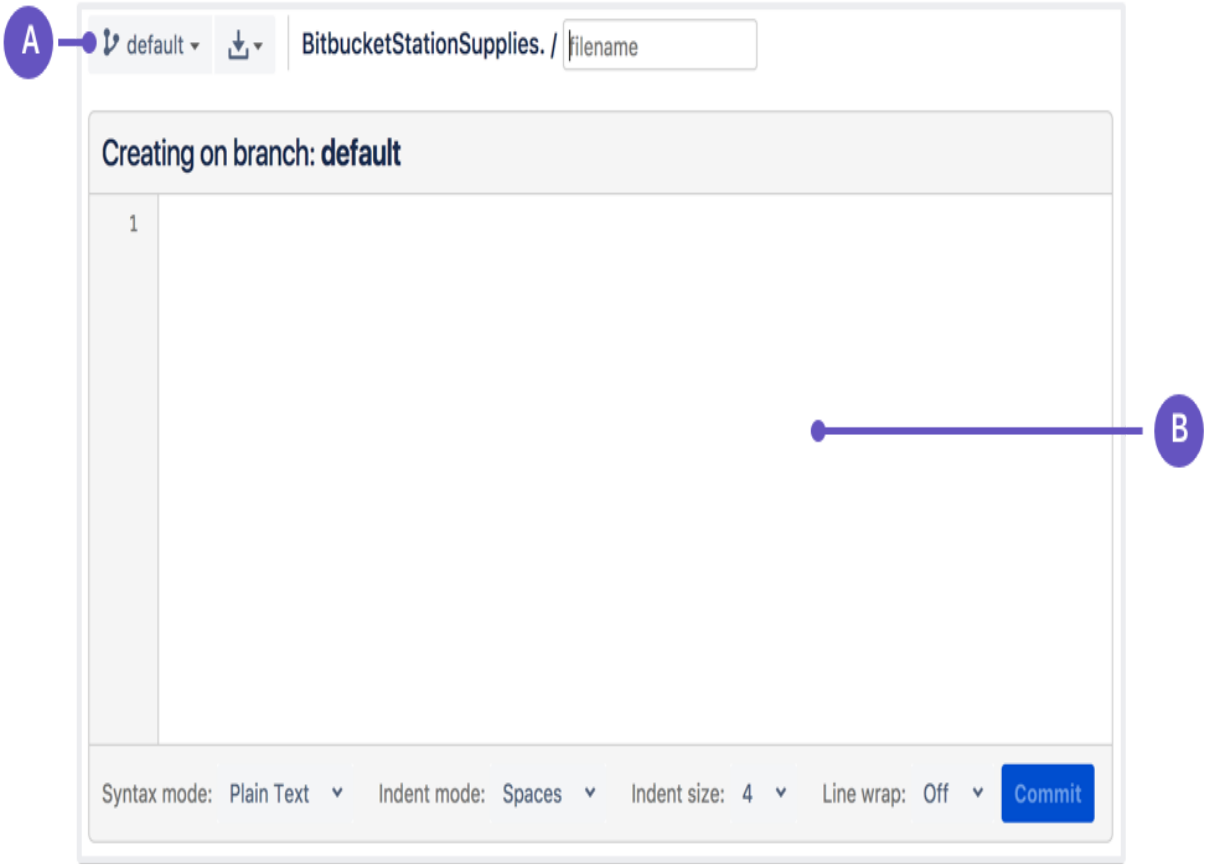
1. From your **BitbucketStationLocations** repository, click **Source** to open the source directory. Notice you only have one file, `locations.txt` , in your directory.



A. Source page: Click the link to open this page.
B. Branch selection: Pick the branch you want to view.
C. More options button: Click to open a menu with more options, such as 'Add file'.
D. Source file area: View the directory of files in Bitbucket.

2. From the **Source** page, click the **More options** button in the top right corner and select **Add file** from the menu. The **More options** button only appears after you have added at least one file to the repository.

   A page for creating the new file opens, as shown in the following image.



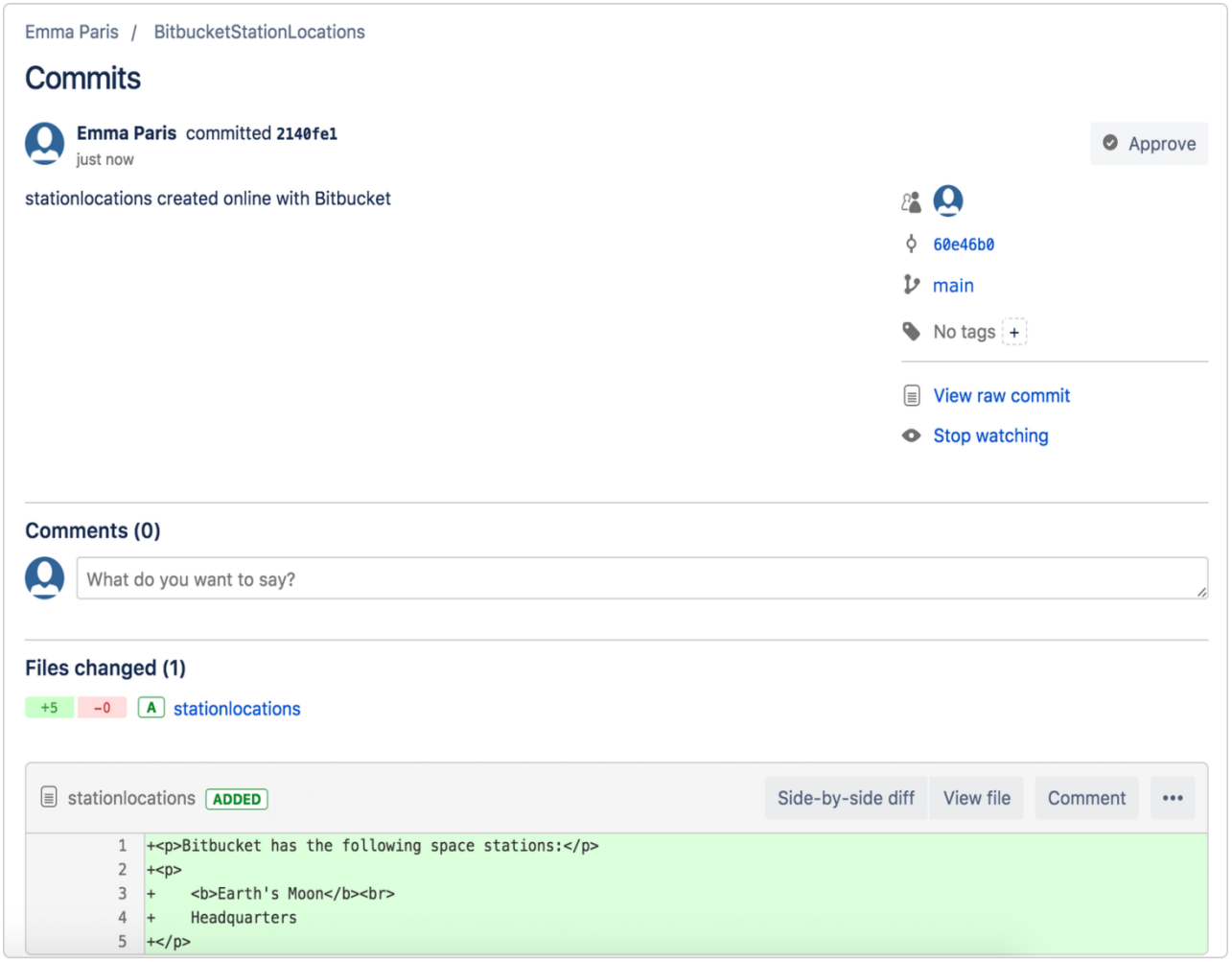
A. Branch with new file: Change if you want to add file to a different branch.
B. New file area: Add content for your new file here.

3. Enter `stationlocations` in the **filename** field.
4. Select **HTML** from the **Syntax mode** list.
5. Add the following HTML code into the text box:

```
<p>Bitbucket has the following space stations:</p>
<p>
    <b>Earth's Moon</b><br>
    Headquarters
</p>
```

6. Click **Commit**. The **Commit message** field appears with the message: `stationlocations created online with Bitbucket.`
7. Click **Commit** under the message field.

You now have a new file in Bitbucket! You are taken to a page with details of the commit, where you can see the change you just made:



If you want to see a list of the commits you've made so far, click **Commits** in the sidebar.

Step 2. Pull changes from a remote repository

Now we need to get that new file into your local repository. The process is pretty straight forward, basically just the reverse of the push you used to get the `locations.txt` file into Bitbucket.

To pull the file into your local repository, do the following:

1. Open your terminal window and navigate to the top level of your local repository.

   ```
   $ cd ~/repos/bitbucketstationlocations/
   ```

2. Enter the `git pull --all` command to pull all the changes from Bitbucket. (In more complex branching workflows, pulling and merging all changes might not be appropriate .) Enter your Bitbucket password when asked for it. Your terminal should look similar to the following

   ```
   $ git pull --all
    Fetching origin
    remote: Counting objects: 3, done.
    remote: Compressing objects: 100% (3/3), done.
    remote: Total 3 (delta 0), reused 0 (delta 0)
    Unpacking objects: 100% (3/3), done.
    From https://bitbucket.org/emmap1/bitbucketstationlocations
       fe5a280..fcbeeb0 main -> origin/main
    Updating fe5a280..fcbeeb0
    Fast-forward
   ```

```
stationlocations | 5 ++++++++++++++
1 file changed, 5 insertions(+)
create mode 100644 stationlocations
```

The `git pull` command merges the file from your remote repository (Bitbucket) into your local repository with a single command.

3. Navigate to your repository folder on your local system and you'll see the file you just added.

Fantastic! With the addition of the two files about your space station location, you have performed the basic Git workflow (clone, add, commit, push, and pull) between Bitbucket and your local system.

GitHub

What is GitHub?

Github is an online platform where we can share our codes(or projects) online hassle-free. Github is place where we host our local git repository online. it basically allow you to work collaboratively within a group of peoples. It hosts all the features of Git and have its own too.

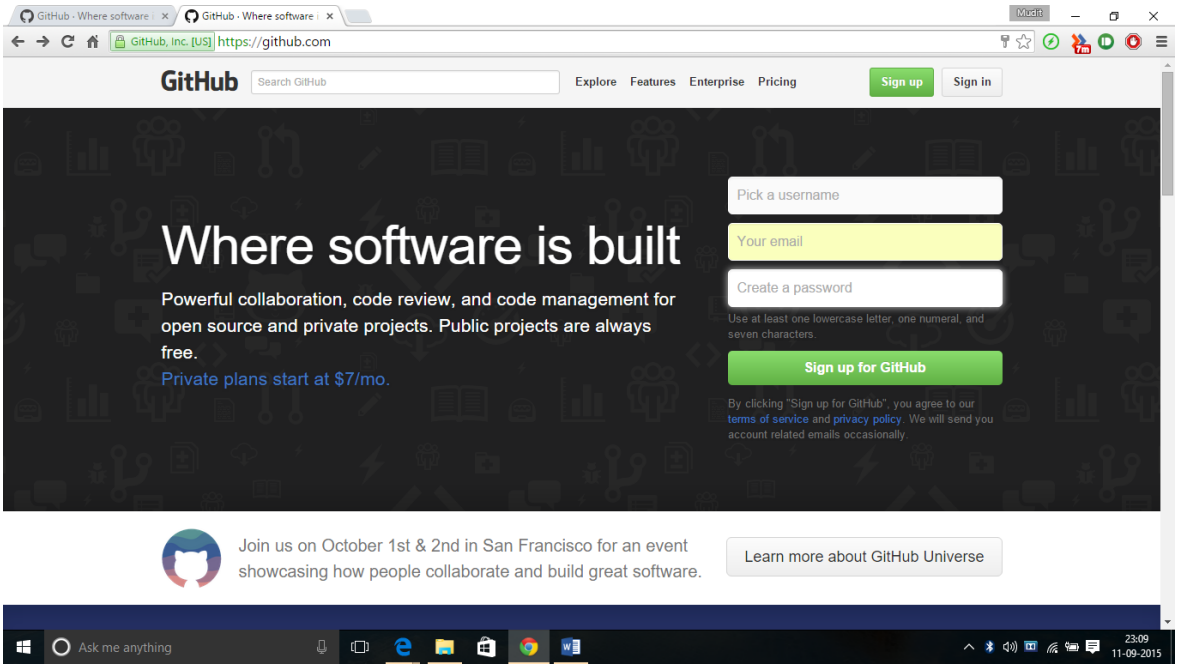
In simple words, Github might be considered as a social media which is made for the developers where they share their work. it might be any project regarding website development or any design of website, or some operating systems like Android, or Linux, etc.

Github is owned by Microsoft, provides access to public(free) and private(paid) repositories. It allows free repository to host files up to 100mb and total size for 2GB.

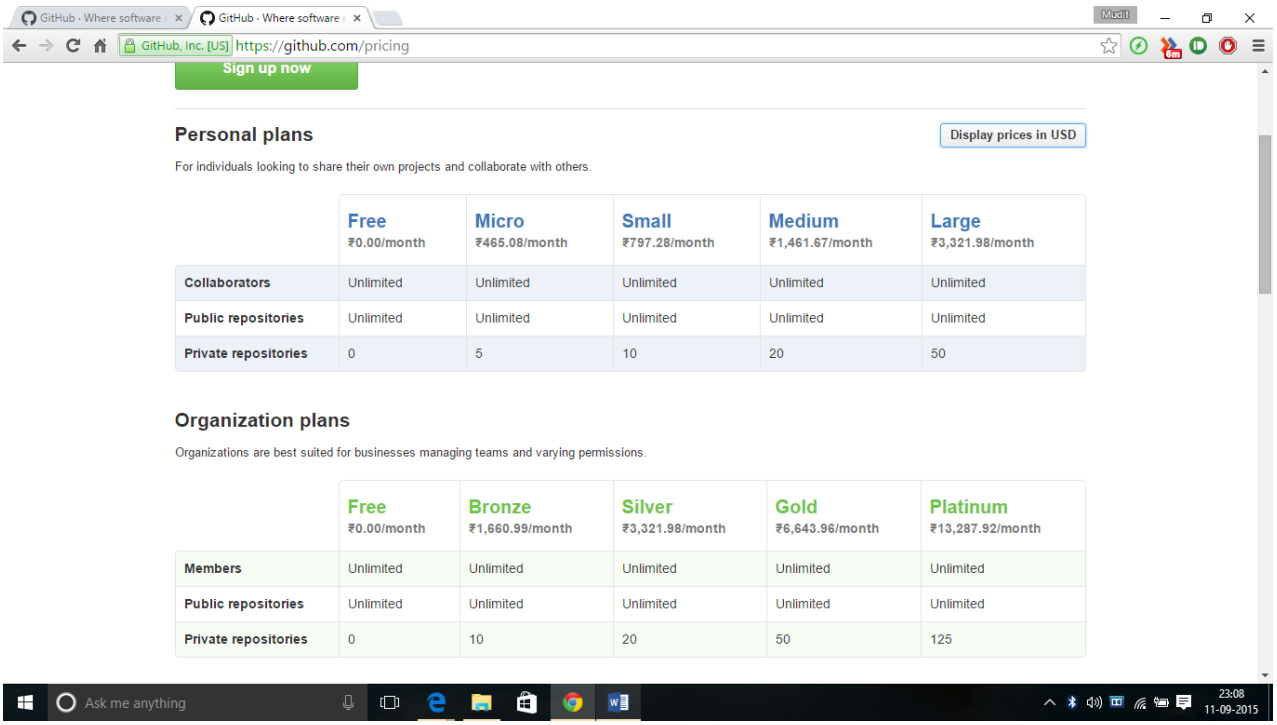
Let's us see how to host to a local repository to Github, from very beginning(creating a github account).

A. Creating a GitHub Account

Step 1: Go to github.com and enter the required user credentials asked on the site and then click on the SignUp for GitHub button.

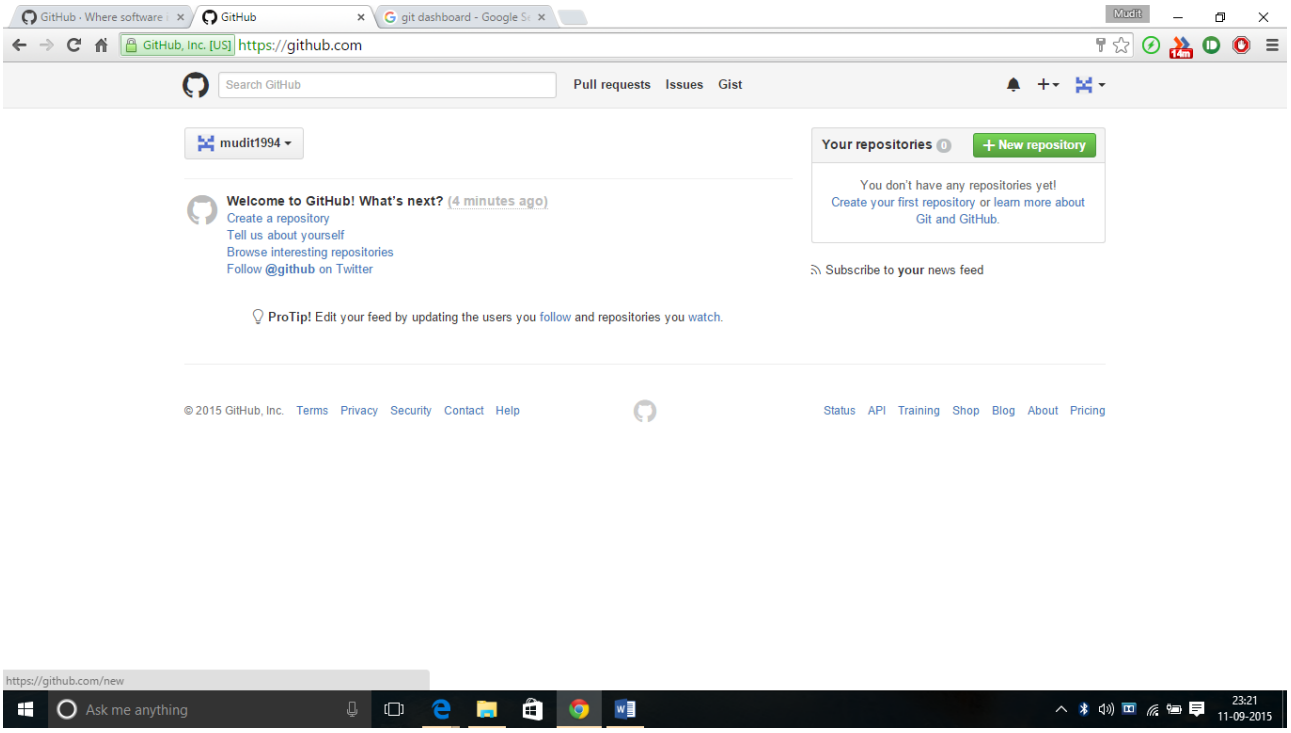


Step 2: Choose a plan that best suits you. The following plans are available as shown in below media as depicted:



Step 3: Then Click on Finish Sign Up.

The account has been created. The user is automatically redirected to your Dashboard.

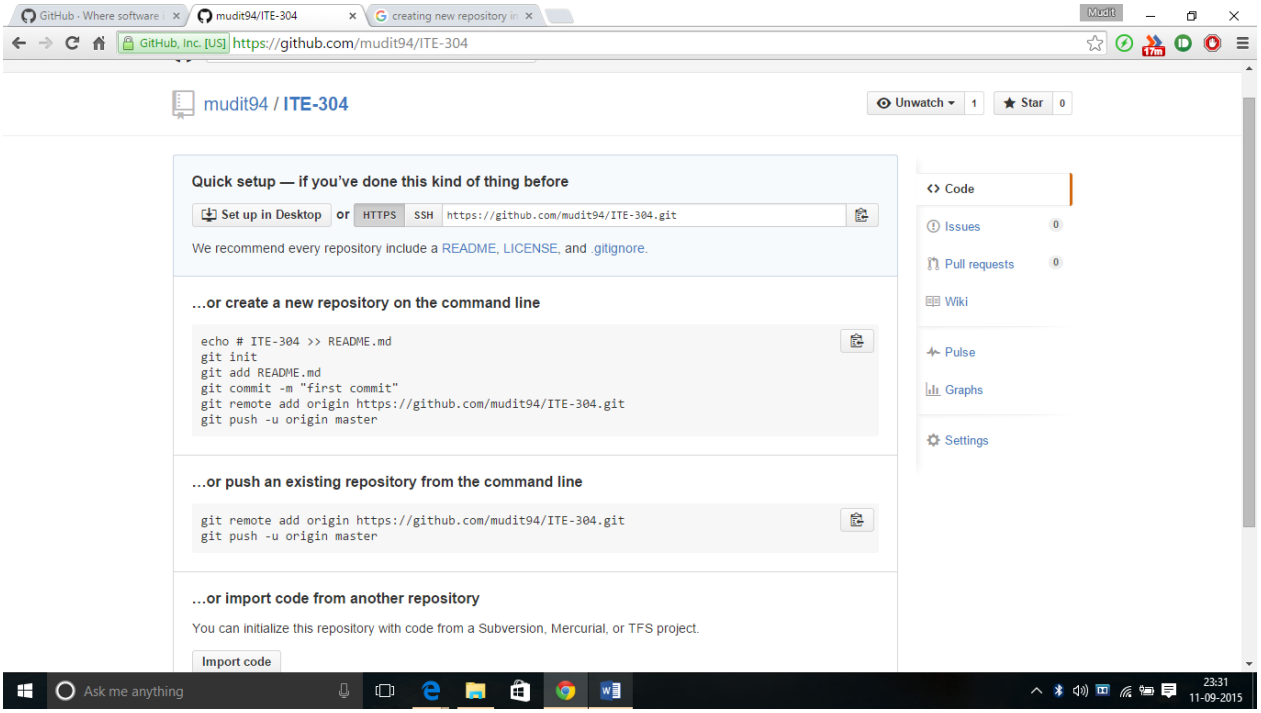


B. Creating a new Repository

- Login to your Github account
- On the dashboard click on the Green Button starting New repository.
- Make sure to **verify the Github account** by going into the mail which was provided when creating the account.

- Once verification has been done, the following screen comes-

C. Start by giving a repository name, description(optional) and select the visibility and accessibility mode for the repository
D. Click on Create repository
E. The repository (in this case ITE-304 is the repository) is now created. The repository can be created looks like:



Uploading Existing Repository to GitHub

- The **system should have git installed** in it if not install git. Make sure to choose Run git from Windows Command prompt option during installation. Otherwise, open git bash in place of step 2.
- Open Terminal (for Mac users) or the command prompt (for Windows and Linux users).
- Change the current working directory to your local project
- Initialize the local directory as a git repository in different ways as described in the image.

```
gitinit
```

- A **new .git folder** is **created** in the directory which is by **default hidden**.
- Add the files in your new local repository. This stages them for the first commit.

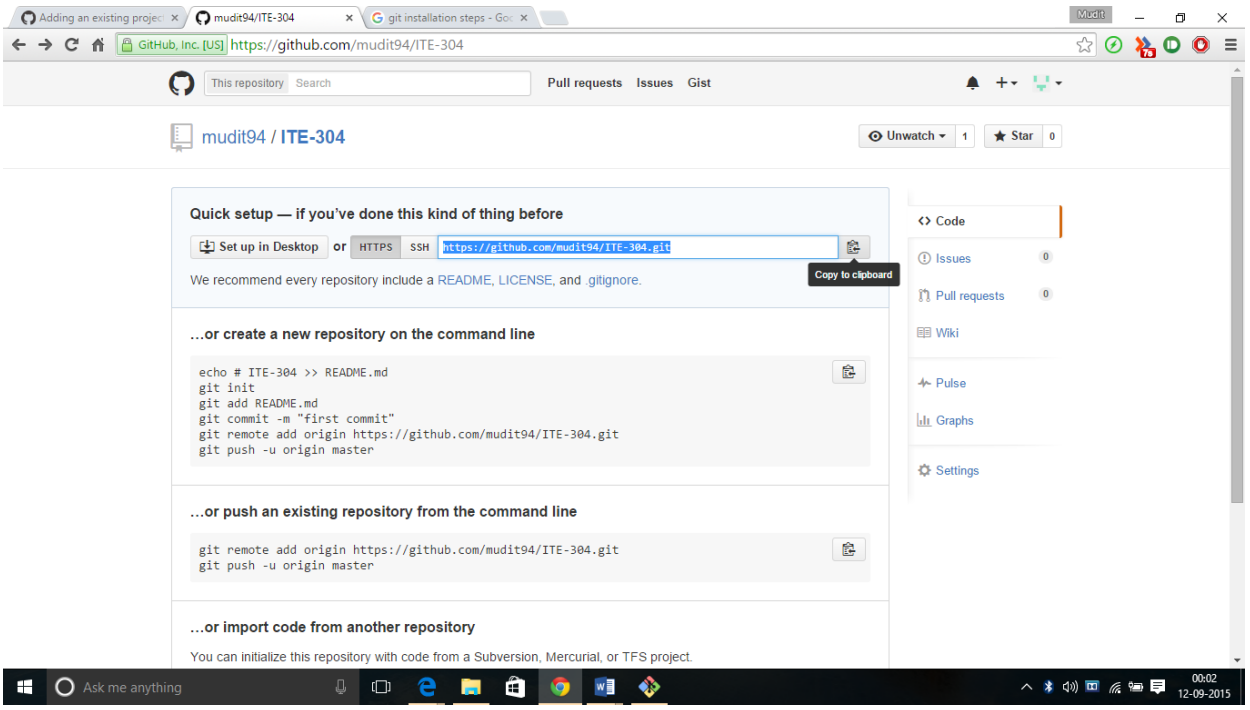
```
git add .
#Adds the files in the local repository and stages them for commit. To unstage a
file, use
git reset HEAD YOUR_FILE
```

Commit the files that you've staged in your local repository.

```
git commit -m 'First commit'
# Commits the tracked changes and prepares them to be pushed to a remote repository.
  To remove this commit and modify the file, use
git reset --soft HEAD~1
```

```
And commit and add the file again.
```

At the top of the GitHub repository's Quick Setup page, click on the icon shown and copy the remote repository URL.



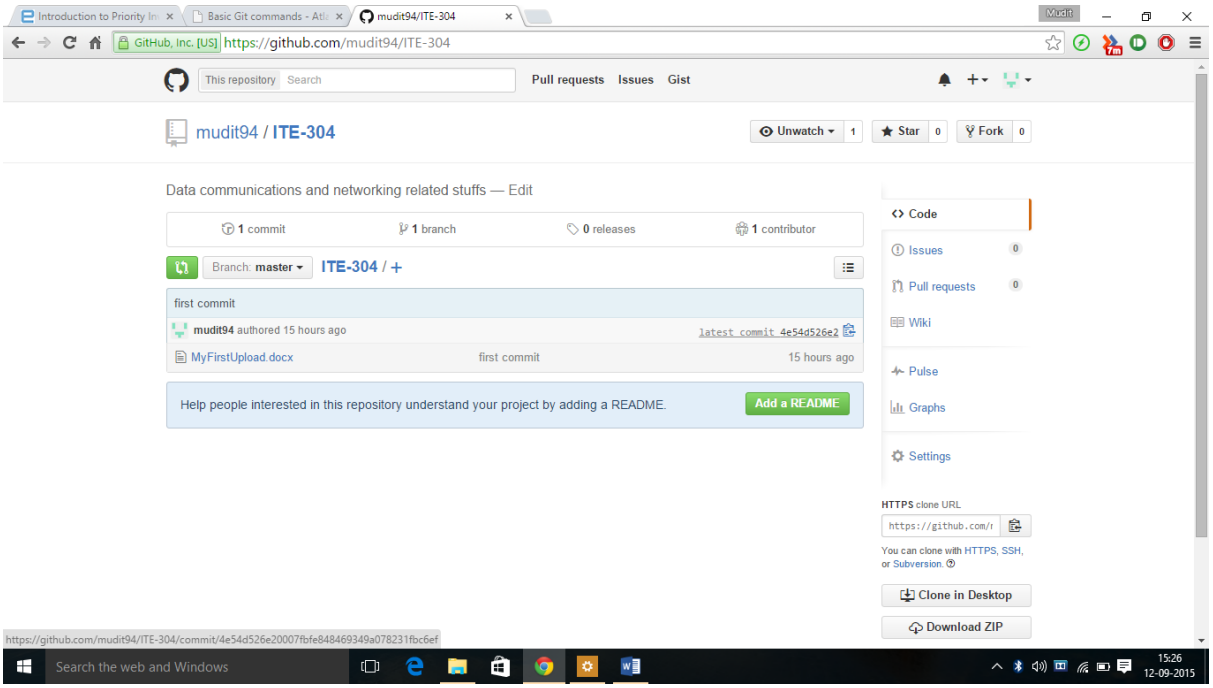
In the Command prompt, **add the URL for the remote repository** where your local repository will be pushed.

```
git remote add origin remote repository URL
 # Connects to the remote repository
git remote -v
 # Verifies the new remote URL
```

Push the changes in your local repository to GitHub.

```
git push origin master
# Pushes the changes in your local repository up to the remote repository you
specified as the origin.
```

And here you go…



You may download changes from remote repository to local repository using the command:

```
git pull
```

- BasicsofdistributedgitAccountcreationandconfiguration
- Createandpushtorepositories
- versioning
- Collaboration
- Migration

Createrepository–namedminiproject-1PushthesametoGitHub

W-3 D-3 FN 3 Hrs P-1Hr

Cloudbasics

CloudInfrastructureOverview

Cloud computing is the delivery of computing resources over the internet. It offers cost savings, scalability, high performance, economies of scale, and more. For many companies, a cloud migration is directly related to data and IT modernization.

What is Cloud-Computing?

Cloud computing is the delivery of computing resources — including storage, processing power, databases, networking, analytics, artificial intelligence, and software applications — over the internet (the cloud). By outsourcing these resources, companies can access the computational assets they need,

when they need them, without needing to purchase and maintain a physical, on-premise IT infrastructure. This provides flexible resources, faster innovation, and economies of scale. For many companies, a cloud migration is directly related to data and IT modernization.

Benefits of cloud computing
The following are the key cloud computing benefits for agile teams.

1.Reduced cost : Teams that use cloud resources do not have to purchase their own hardware assets. Beyond hardware costs, cloud providers do their best to maximize and optimize hardware usage.
2.**Increased scalability :** Cloud applications can automatically shrink and grow their infrastructure resources in response to spikes of traffic.
3.Better performance : Cloud computing offers the latest and greatest computational resources. Users can access the newest machines with extreme, multi-core CPUs designed for heavy parallel processing tasks.
4.Improved execution speed : Teams that use cloud infrastructures can more rapidly execute and deliver value to their customers. Agile software teams can leverage a cloud infrastructure to rapidly spin up new virtual machines to experiment and validate unique ideas, and automate the testing and deployment phases of the pipeline.

5.Increased security : Cloud providers offer many security mechanisms and technologies to help build secure applications. User access control is an important security concern, and most cloud providers offer tools to limit granular user access.
6.Continuous integration and delivery : Continuous integration and continuous delivery (CI/CD) is a key practice for DevOps practitioners that helps to increase team velocity and reduce time to market. Cloud-based CI/CD, such as Bitbucket Pipelines, allows teams to automatically build, test and deploy code. **7.Comprehensive monitoring and incident management** Comprehensive monitoring is another key capability for organizations practicing DevOps because it allows them to address issues and incidents faster. Cloud providers share metrics about the health of the system, including application and server CPU, memory, request rate, error rate, average response time, etc.

Cloudcomputingarchitectureanditscomponents

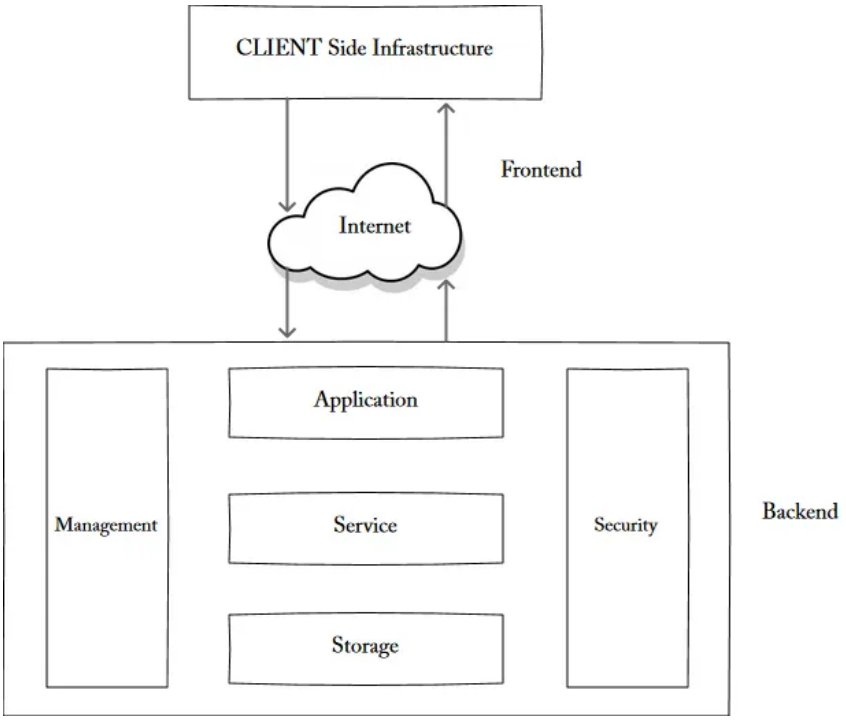
The cloud infrastructure is closely related to its architecture and comprises many cloud components that are loosely connected.

The broad divisions of cloud architecture are:

- Front-end
- Back-end

It is the **back-end** responsibility to provide data security for cloud users and the traffic control mechanism. The server also provides the middleware, which helps to connect devices and communicate with each other.

Figure - Cloud Computing Architecture:



The cloud technology architecture also consists of **front-end** platforms (as read in the early chapters) called the cloud client, which comprises servers, thin & fat clients, tablets & mobile devices. The interaction is done through middleware or via web-browser or virtual sessions. According to Jason Bloomberg of ZapThink, the cloud-oriented architecture can essentially be the building block of IoT (Internet of Things) in which anything can be connected to the internet. Cloud architecture is a combination of both services-oriented architecture & event-driven architecture. SO cloud architecture encompasses all elements of the cloud environment.

Servicemodels

The cloud computing service models are categorized into three different types:

1. Infrastructure as a Service (IaaS)
2. Platform as a Service (PaaS)
3. Software as a Service (SaaS)

1.Infrastructure as a Service (IaaS)

IaaS is also known as **Hardware as a Service (HaaS)**. It is a computing infrastructure managed over the internet. The main advantage of using IaaS is that it helps users to avoid the cost and complexity of purchasing and managing the physical servers.

Characteristics of IaaS

There are the following characteristics of IaaS -

- Resources are available as a service
- Services are highly scalable
- Dynamic and flexible
- GUI and API-based access
- Automated administrative tasks

Example: DigitalOcean, Linode, Amazon Web Services (AWS), Microsoft Azure, Google Compute Engine (GCE), Rackspace, and Cisco Metacloud.

2.Platform as a Service (PaaS)
PaaS cloud computing platform is created for the programmer to develop, test, run, and manage the applications.

Characteristics of PaaS
There are the following characteristics of PaaS -

- Accessible to various users via the same development application.
- Integrates with web services and databases.
- Builds on virtualization technology, so resources can easily be scaled up or down as per the organization's need.
- Support multiple languages and frameworks.
- Provides an ability to "**Auto-scale**".

Example: AWS Elastic Beanstalk, Windows Azure, Heroku, Force.com, Google App Engine, Apache Stratos, Magento Commerce Cloud, and OpenShift.

3.Software as a Service (SaaS)

SaaS is also known as "**on-demand software**". It is a software in which the applications are hosted by a cloud service provider. Users can access these applications with the help of internet connection and web browser.

Characteristics of SaaS
There are the following characteristics of SaaS -

- Managed from a central location
- Hosted on a remote server
- Accessible over the internet
- Users are not responsible for hardware and software updates. Updates are applied automatically.
- The services are purchased on the pay-as-per-use basis

Example: BigCommerce, Google Apps, Salesforce, Dropbox, ZenDesk, Cisco WebEx, ZenDesk, Slack, and GoToMeeting.

Difference between IaaS, PaaS, and SaaS

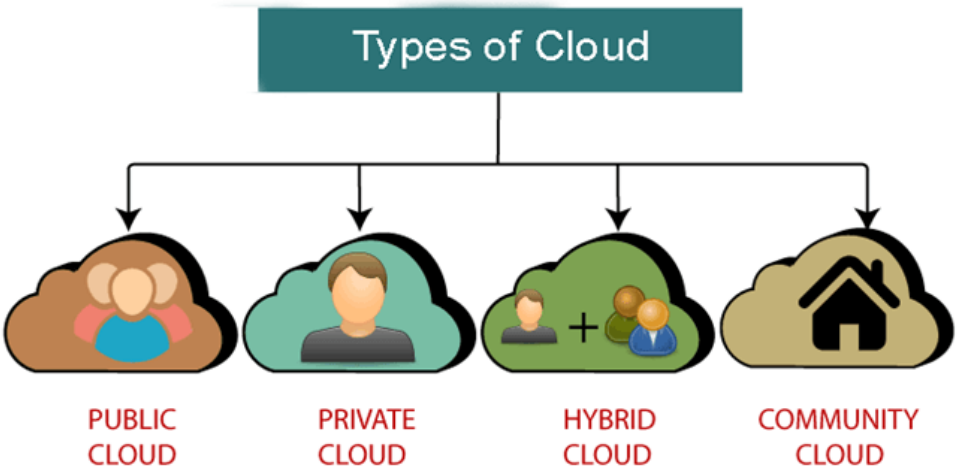
The below table shows the difference between IaaS, PaaS, and SaaS -

IaaS	Paas	SaaS
It provides a virtual data center to store information and create platforms for app development, testing, and deployment.	It provides virtual platforms and tools to create, test, and deploy apps.	It provides web software and apps to complete business tasks.
It provides access to resources such as virtual machines, virtual storage, etc.	It provides runtime environments and deployment tools for applications.	It provides software as a service to the end-users.
It is used by network architects.	It is used by developers.	It is used by end users.
IaaS provides only Infrastructure.	PaaS provides Infrastructure+Platform.	SaaS provides Infrastructure+Platform +Software.

Deployment models

There are 4 types of cloud that you can deploy according to the organization's needs-

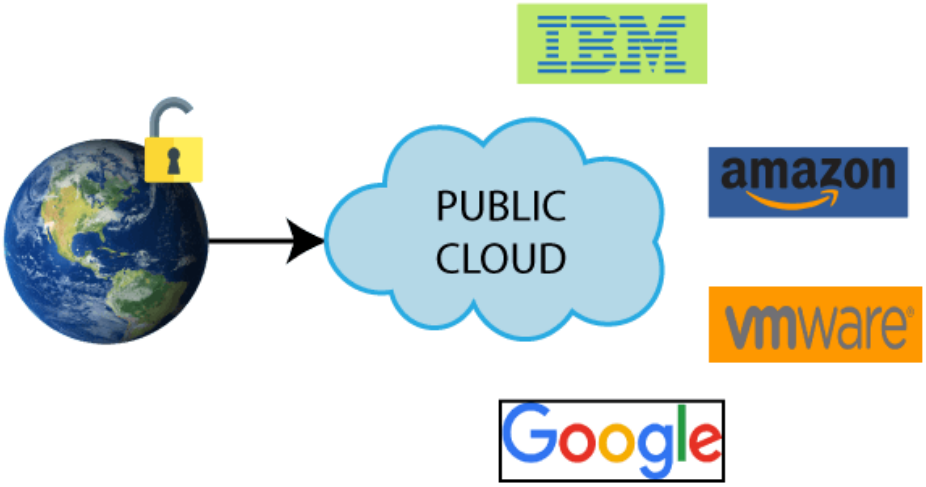
1. Public Cloud
2. Private Cloud
3. Hybrid Cloud
4. Community Cloud



1.Public Cloud

Public cloud is **open to all** to store and access information via the Internet using the pay-per-usage method.In public cloud, computing resources are managed and operated by the Cloud Service Provider (CSP).

Example: Amazon elastic compute cloud (EC2), IBM SmartCloud Enterprise, Microsoft, Google App Engine, Windows Azure Services Platform.



Advantages of Public Cloud

There are the following advantages of Public Cloud -

- Public cloud is owned at a lower cost than the private and hybrid cloud.
- Public cloud is maintained by the cloud service provider, so do not need to worry about the maintenance.
- Public cloud is easier to integrate. Hence it offers a better flexibility approach to consumers.

- Public cloud is location independent because its services are delivered through the internet.
- Public cloud is highly scalable as per the requirement of computing resources.
- It is accessible by the general public, so there is no limit to the number of users.

Disadvantages of Public Cloud

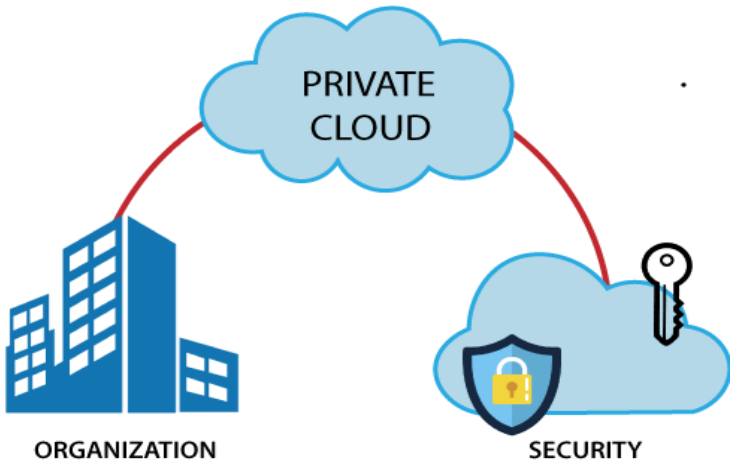
- Public Cloud is less secure because resources are shared publicly.
- Performance depends upon the high-speed internet network link to the cloud provider.
- The Client has no control of data.

2. Private Cloud

Private cloud is also known as an **internal cloud** or **corporate cloud**. It is used by organizations to build and manage their own data centers internally or by the third party. It can be deployed using Opensource tools such as Openstack and Eucalyptus.

Based on the location and management, National Institute of Standards and Technology (NIST) divide private cloud into the following two parts-

- On-premise private cloud
- Outsourced private cloud



Advantages of Private Cloud

There are the following advantages of the Private Cloud -

- Private cloud provides a high level of security and privacy to the users.
- Private cloud offers better performance with improved speed and space capacity.
- It allows the IT team to quickly allocate and deliver on-demand IT resources.
- The organization has full control over the cloud because it is managed by the organization itself. So, there is no need for the organization to depends on anybody.
- It is suitable for organizations that require a separate cloud for their personal use and data security is the first priority.

Disadvantages of Private Cloud

- Skilled people are required to manage and operate cloud services.
- Private cloud is accessible within the organization, so the area of operations is limited.
- Private cloud is not suitable for organizations that have a high user base, and organizations that do not have the prebuilt infrastructure, sufficient manpower to maintain and manage the cloud.

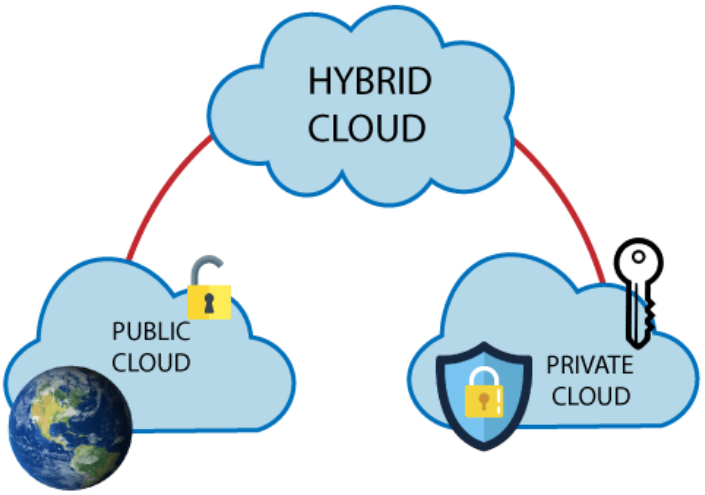
3.Hybrid Cloud

Hybrid Cloud is a combination of the public cloud and the private cloud. we can say:

Hybrid Cloud = Public Cloud + Private Cloud

Hybrid cloud is partially secure because the services which are running on the public cloud can be accessed by anyone, while the services which are running on a private cloud can be accessed only by the organization's users.

Example: Google Application Suite (Gmail, Google Apps, and Google Drive), Office 365 (MS Office on the Web and One Drive), Amazon Web Services.



Advantages of Hybrid Cloud

There are the following advantages of Hybrid Cloud -

- Hybrid cloud is suitable for organizations that require more security than the public cloud.
- Hybrid cloud helps you to deliver new products and services more quickly.
- Hybrid cloud provides an excellent way to reduce the risk.
- Hybrid cloud offers flexible resources because of the public cloud and secure resources because of the private cloud.

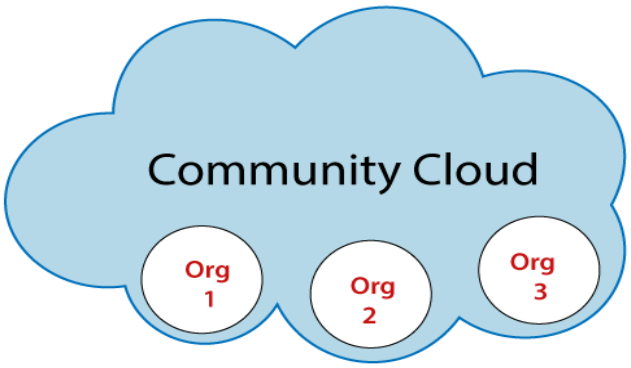
Disadvantages of Hybrid Cloud

- In Hybrid Cloud, security feature is not as good as the private cloud.
- Managing a hybrid cloud is complex because it is difficult to manage more than one type of deployment model.
- In the hybrid cloud, the reliability of the services depends on cloud service providers.

4.Community Cloud

Community cloud allows systems and services to be accessible by a group of several organizations to share the information between the organization and a specific community. It is owned, managed, and operated by one or more organizations in the community, a third party, or a combination of them.

Example: Health Care community cloud



Advantages of Community Cloud

There are the following advantages of Community Cloud -

- Community cloud is cost-effective because the whole cloud is being shared by several organizations or communities.
- Community cloud is suitable for organizations that want to have a collaborative cloud with more security features than the public cloud.
- It provides better security than the public cloud.
- It provdes collaborative and distributive environment.
- Community cloud allows us to share cloud resources, infrastructure, and other capabilities among various organizations.

Disadvantages of Community Cloud

- Community cloud is not a good choice for every organization.
- Security features are not as good as the private cloud.
- It is not suitable if there is no collaboration.
- The fixed amount of data storage and bandwidth is shared among all community members.

Difference between public cloud, private cloud, hybrid cloud, and community cloud -

The below table shows the difference between public cloud, private cloud, hybrid cloud, and community cloud.

Parameter	Public Cloud	Private Cloud	Hybrid Cloud	Community Cloud
Host	Service provider	Enterprise (Third party)	Enterprise (Third party)	Community (Third party)
Users	General public	Selected users	Selected users	Community members
Access	Internet	Internet, VPN	Internet, VPN	Internet, VPN
Owner	Service provider	Enterprise	Enterprise	Community

Virtualization

Virtualization is the "creation of a virtual (rather than actual) version of something, such as a server, a desktop, a storage device, an operating system or network resources".

In other words, Virtualization is a technique, which allows to share a single physical instance of a resource or an application among multiple customers and organizations. It does by assigning a logical name to a physical storage and providing a pointer to that physical resource when demanded.

What is the concept behind the Virtualization?

Creation of a virtual machine over existing operating system and hardware is known as Hardware Virtualization. A Virtual machine provides an environment that is logically separated from the underlying hardware.

The machine on which the virtual machine is going to create is known as **Host Machine** and that virtual machine is referred as a **Guest Machine**

Types of Virtualization:

1. Hardware Virtualization.
2. Operating system Virtualization.
3. Server Virtualization.
4. Storage Virtualization.
5. Data virtualization

1) Hardware Virtualization:

When the virtual machine software or virtual machine manager *(VMM) is directly installed on the hardware system* is known as hardware virtualization.

The main job of hypervisor is to control and monitoring the processor, memory and other hardware resources.

After virtualization of hardware system we can install different operating system on it and run different applications on those OS.

Usage:

Hardware virtualization is mainly done for the server platforms, because controlling virtual machines is much easier than controlling a physical server.

2) Operating System Virtualization:

When the virtual machine software or virtual machine manager *(VMM) is installed on the Host operating system* instead of directly on the hardware system is known as operating system virtualization.

Usage:

Operating System Virtualization is mainly used for testing the applications on different platforms of OS.

3) Server Virtualization:

When the virtual machine software or virtual machine manager *(VMM) is directly installed on the Server system* is known as server virtualization.

Usage:

Server virtualization is done because a single physical server can be divided into multiple servers on the demand basis and for balancing the load.

4) Storage Virtualization:

Storage virtualization is the *process of grouping the physical storage from multiple network storage devices so that it looks like a single storage device*.

Storage virtualization is also implemented by using software applications.

Usage: Storage virtualization is mainly done for back-up and recovery purposes.

5) Data virtualization

Data virtualization is the process of retrieve data from various resources without knowing its type and physical location where it is stored.

Usage:

It collects heterogeneous data from different resources and allows data users across the organization to access this data according to their work requirements. This heterogeneous data can be accessed using any application such as web portals, web services, E-commerce, Software as a Service (SaaS), and mobile application.

Cloud Native Application Development

What is a cloud-native application?

A cloud-native application is a program that is designed for a cloud computing architecture. A native app is software that is developed for use on a specific platform or device.Cloud-native applications use a microservice architecture..

Features of a cloud-native application

The microservices that are part of the cloud-native app architecture are packaged in containers that connect and communicate via APIs.

Here are some of the key features or capabilities of these applications:

- **Microservices-based.**Microservices break down an application into a series of independent services, or modules. These modules communicate with one another via application program interfaces (APIs).
- **Container-based.** Containers keep microservices from interfering with one another. They also enable multiple instances of the same service.
- **API-based.** APIs connect microservices and containers while providing simplified maintenance and security.
- **Dynamically orchestrated.** Container orchestration tools handle resource management, load balancing, scheduling of restarts after an internal failure and provisioning and deploying containers onto server cluster nodes.

Benefits of cloud-native applications

Cloud-native applications are designed to take advantage of the speed and efficiency of the cloud. Some of the benefitsof cloud-native applications are:

- **Cost-effective.** Computing and storage resources can scale out as needed. Containers can also be used to maximize the number of microservices run on a host, saving time, resources and money.
- **Independently scalable.** Each microservice is logically isolated and can scale independently.
- **Portability.** Cloud-native applications are vendor neutral and use containers to port microservices between different vendors' infrastructure, helping avoid vendor lock-in.
- **Reliable.** If a failure occurs in one microservice, there's no effect on adjacent services because these cloud-based applications use containers.
- **Easy to manage.** Cloud-native applications use automation to deploy app features and updates. Developers can track all microservices and components as they're being updated..
- **Visibility.** Because a microservice architecture isolates services, it makes it easier for engineering teams to study applications and learn how they function together.

Cloud-native application development

Best practices for designing cloud-native applications are based on the DevOps principle All cloud-native application designs should consider how the app will be built, how performance is measured and how teams foster continuous improvement through the app lifecycle. Here are the five parts of design:

1. **Automate.** Automation allows for the consistent provisioning of cloud application environments across multiple cloud vendors. With automation, infrastructure as code (IaC) is used to track changes in a source code repository.
2. **Monitor.** Teams should monitor the development environment, as well as how the application is being used..
3. **Document.** Documentation is important to track changes and see how each team is contributing to the application.
4. **Make incremental changes.** Any changes made to the application or the underlying architecture should be incremental and reversible. This will enable teams to learn from changes and not make permanent mistakes. With IaC, developers can track changes in a source repository.
5. **Design for failure.** Processes should be designed for when things inevitably go wrong in a cloud environment. This means implementing test frameworks to simulate failures and learn from them.

Tools for cloud-native app development

Several software tools are used for each cloud-native application development process. Together, they create a development stack.Here is the software found in a cloud-native development stack:

Docker. The Docker platform is open source. It creates, deploys and manages virtualized application containers using a common operating system (OS). It isolates resources allowing multiple containers to use the same OS without contention.

Kubernetes. The Kubernetesplatform is used to manage and orchestrate Linux containers, determining how and where the containers will run.

Terraform. Designed for implementing IaC, Terraformdefines resources as code and applies version control so users can see when and where resources were altered.

GitLab CI/CD. This continuous integration/continuous development (CI/CD) software lets users automate software testing and deployment. GitLab can be used for security analysis, static analysis and unit tests.

Node.js. This JavaScriptruntime is useful for creating real-time applications like chat, news feeds and other microservices. For example, Node.js can create virtual servers and define the routes that connect microservices to external APIs.

W-3 D-3 AN P-3 Hrs

1.Create cloud account (AWS, GCB or anyother service provider) and explore thefeatures.

Sign up using your email address

1. Open the Amazon Web Services (AWS) home page.(https://aws.amazon.com/free/)
2. Choose **Create an AWS Account**.
3. In **Root user email address**, enter your email address, edit the AWS account name, and then choose **Verify email address**. An AWS verification email will be sent to this address with a verification code.

Verify your email address

Enter the code you receive, and then choose **Verify**. The code might take a few minutes to arrive. Check your email and spam folder for the verification code email.

Create your password

Enter your **Root user password** and **Confirm root user password**, and then choose **Continue**.

Add your contact information

1. Select **Personal** or **Business**.
   Note: Personal accounts and business accounts have the same features and functions.

2. Enter your personal or business information.
   Important: For business AWS accounts, it's a best practice to enter the company phone number rather than a personal cell phone number. Configuring a root account with an individual email address or a personal phone number can make your account insecure.

3. Read and accept the AWS Customer Agreement.

4. Choose **Continue**.

You receive an email to confirm that your account is created. You can sign in to your new account using the email address and password that you registered with. However, you can't use AWS services until you finish activating your account.

Add a payment method

On the **Billing information** page, enter the information about your payment method, and then choose **Verify and Add**.

If you are signing up in India for an Amazon Internet Services Private Limited (AISPL) account, then 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 two Indian Rupees (INR), as part of the verification process. AISPL refunds the two INR after the verification is complete.

If you want to use a different billing address for your AWS billing information, choose **Use a new address**. Then, choose **Verify and Continue**.

Important: You can't proceed with the sign-up process until you add a valid payment method.

Verify your phone number

1. On the **Confirm your identity** page, select a contact method to receive a verification code.
2. Select your phone number country or region code from the list.
3. Enter a mobile phone number where you can be reached in the next few minutes.
4. If presented with a CAPTCHA, enter the displayed code, and then submit.
5. In a few moments, an automated system contacts you.
6. Enter the PIN you receive, and then choose **Continue**.

Choose an AWS Support plan

On the **Select a support plan** page, choose one of the available Support plans. For a description of the available Support plans and their benefits, see Compare AWS Support plans.

Choose **Complete sign up**.

Wait for account activation

After you choose a Support plan, a confirmation page indicates that your account is being activated. Accounts are usually activated within a few minutes, but the process might take up to 24 hours.

You can sign in to your AWS account during this time. The AWS home page might display a **Complete Sign Up** button during this time, even if you've completed all the steps in the sign-up process.

When your account is fully activated, you receive a confirmation email. Check your email and spam folder for the confirmation email. After you receive this email, you have full access to all AWS services.

2.**Createandsetupavirtualmachine on AWS.**

What is a virtual machine?

A **virtual machine** (VM) is a software demonstration and development model that simulates an operating system and its applications. VMs create a software simulation environment for testing software and application programs without physical hardware investment.

A virtual machine mimics the different aspects of an operating system or application program with guest operating systems hosted on it. These guest systems can be different architectures (e.g., x86 host and ARM guest) or different versions (e.g., 32-bit Windows XP, hosted within 64-bit Windows 7).

A virtual machine is like any other computer that operates on a central processor and runs guest operating systems. It provides resources beyond a single physical server, such as thousands of servers in terms of CPU power and network bandwidth. VMs are used in cloud computing applications because they allow easier scaling.

Properties of a virtual machine

The common properties of a VM are:

- **Quick operations**: Share system resources and run multiple operating systems instantly
- **Hardware independence**: Setup or migrate to any virtual machine or real server
- **End-to-end isolation**: Improve performance and security with advanced resource controls

Benefits of using a virtual machine

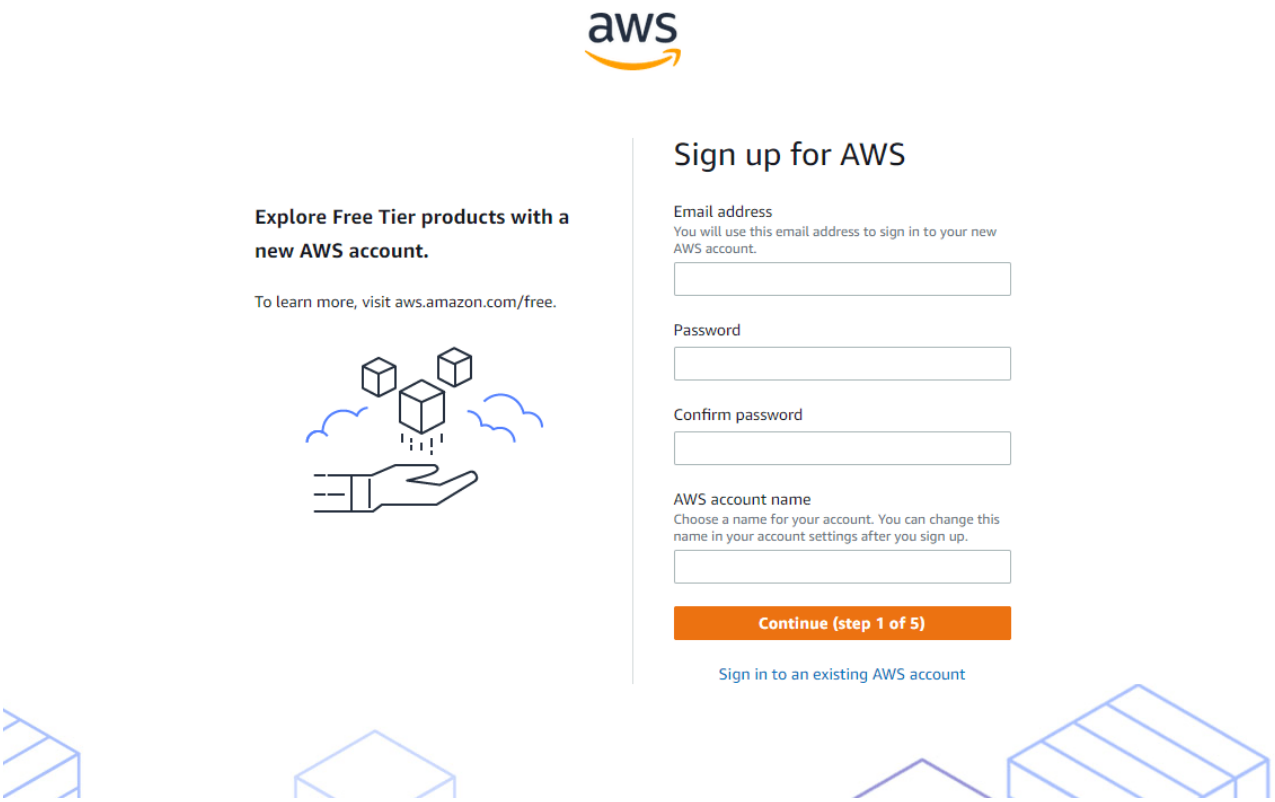
- **Data safety and security:** VMs run in a secure environment and reduce data loss risk. The data is isolated from the host operating system, so it can be copied to another location without spreading across the network.
- **Simplified maintenance:** A virtualization software enables a system administrator to configure a VM's operating system, allocate memory and storage, install applications, and connect to networks.
- **Remote access:** An administrator can access guest operating systems from a central location using graphical management tools and standard network protocols, such as HTTP/HTTPS or RDP/VNC. You can also connect to any VM remotely via a secure shell (SSH) connection. This remote connection provides an easy way for remote support.
- **Flexible resource allocation:** A VM can be configured for optimal resource utilization like CPU power and storage space by creating multiple virtual machines within a single physical server or server cluster.

3. 8 steps to creating a virtual machine with AWS

Amazon Web Services provides scalable and affordable cloud computing services. You can join this platform for free and only pay for what you use. Use the following eight steps to set up a virtual machine with AWS.

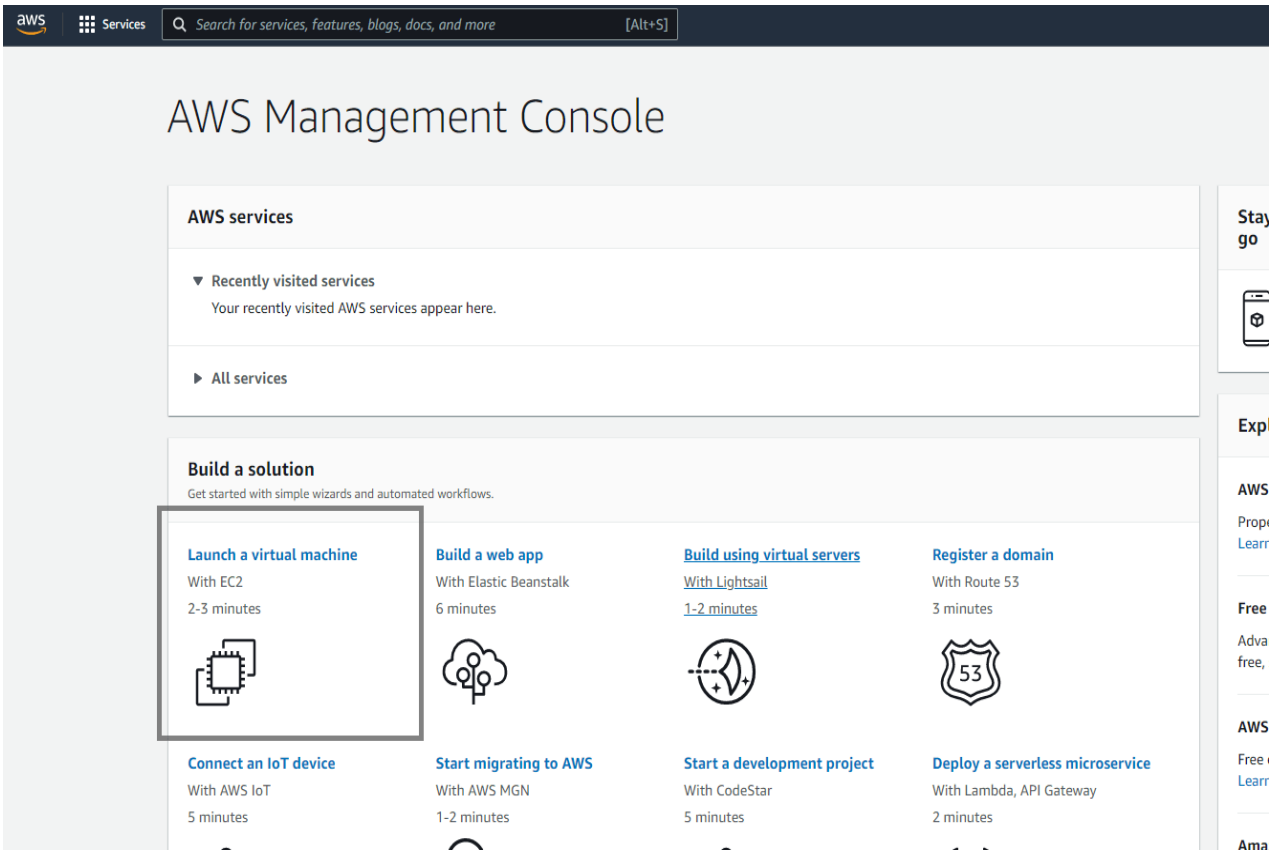
1. Create an AWS account

You can easily create an AWS account on the AWS Console. All new sign-ups get a free-tier offer.



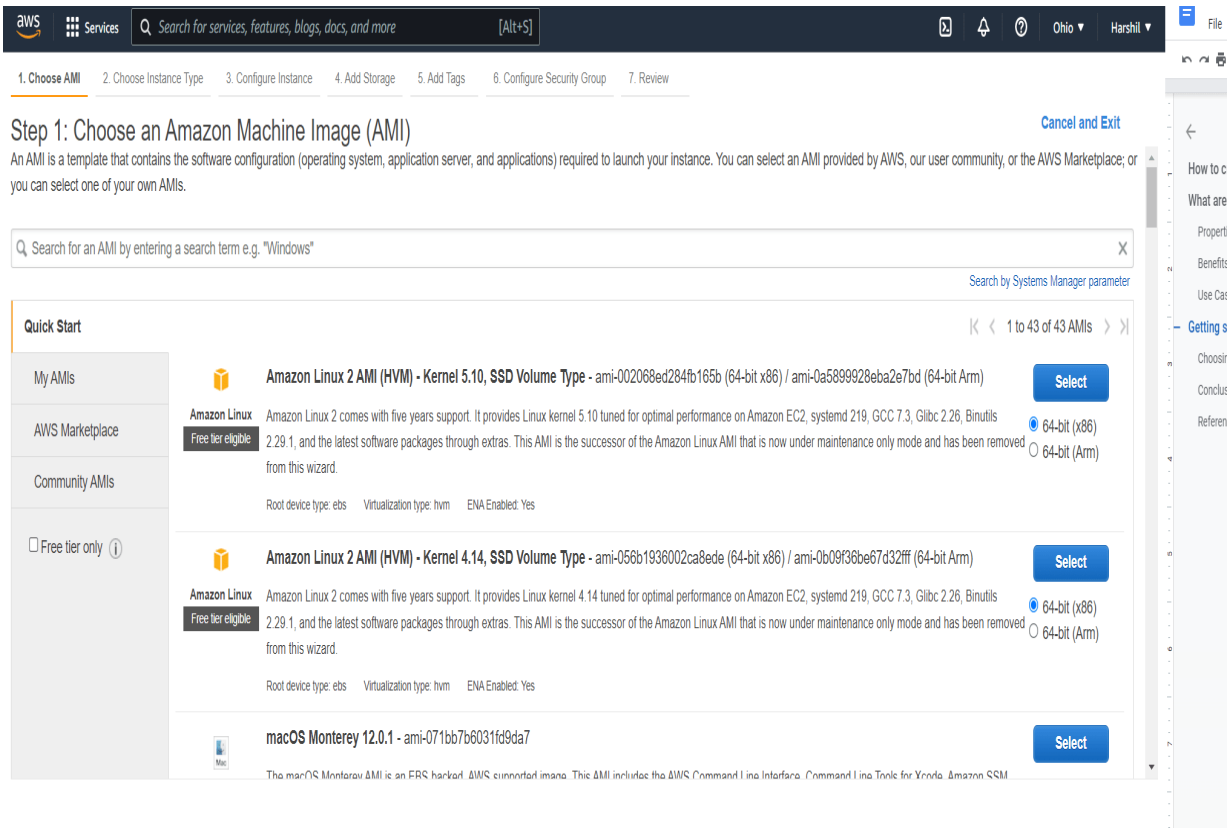
2. Launch AWS virtual machine

Once you finish setting up your account, you can click on the AWS logo on the top left corner or search "console" on the search bar. You'll find a number of options in the AWS console. Select "**Launch a virtual machine**" to get started with VMs. If you're a new user, it can take up to 24 hours for your account to activate.



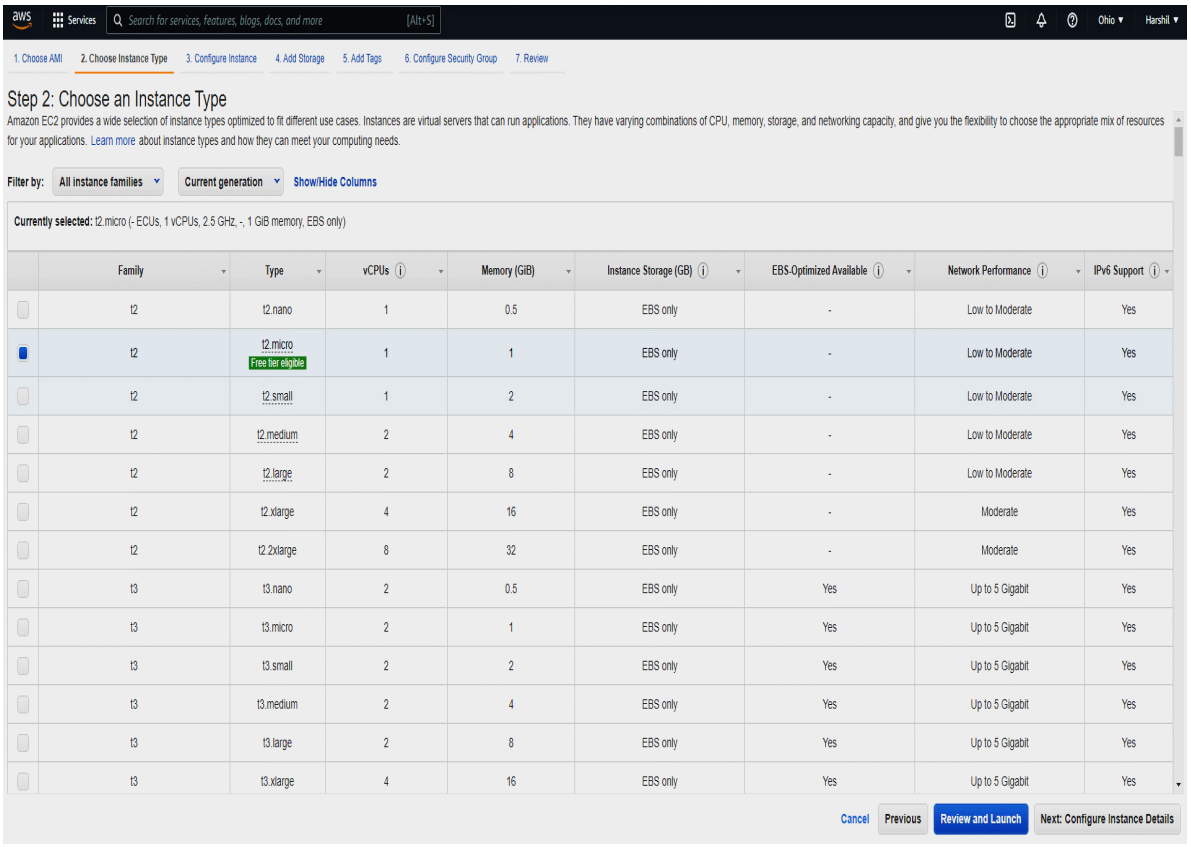
3. Choose AMI

Amazon Machine Image (AMI) highlights the software setup (OS, application server, and apps). You can select Mac, Linux, or Windows OS. We'll look at the setup for Windows virtual machines here.

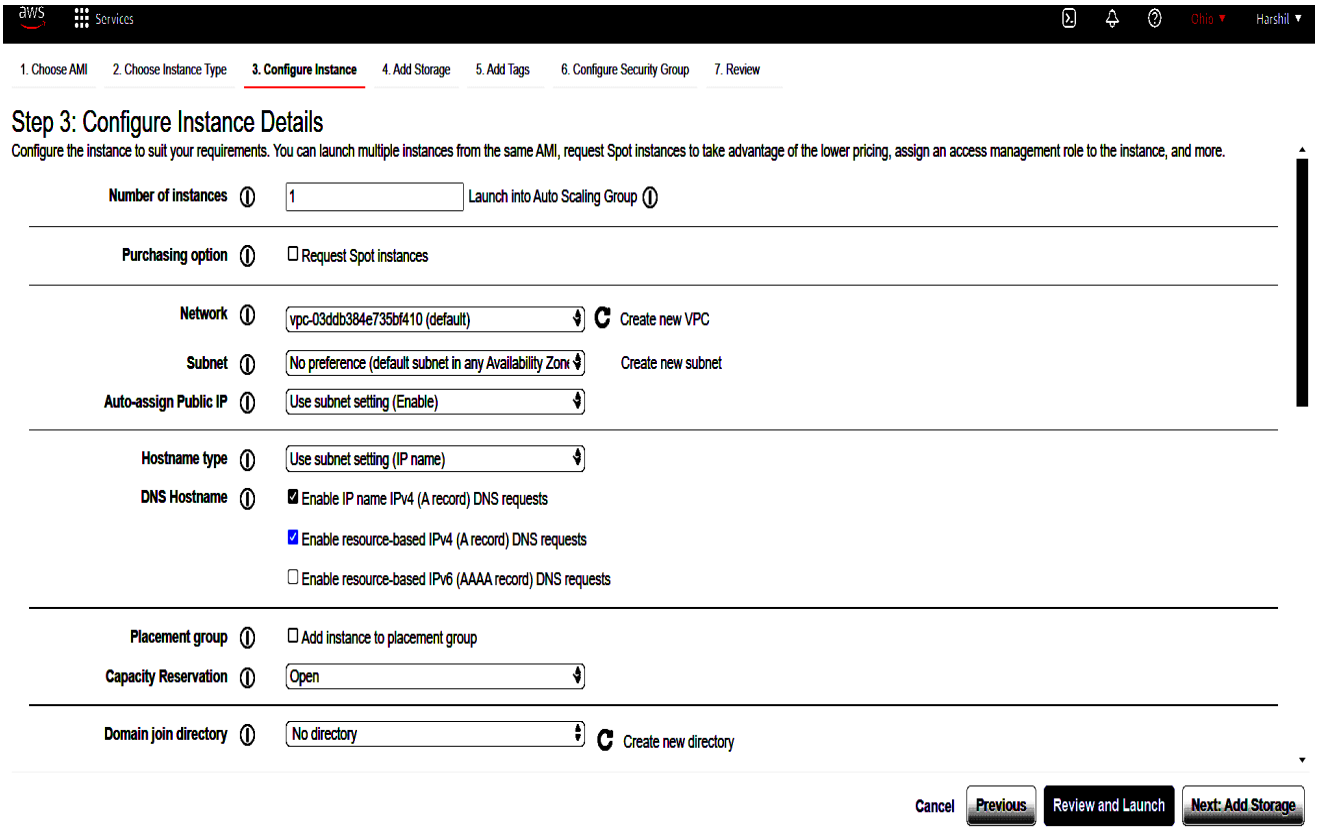


4. Choose and configure instance type

After choosing your operating system, you need to pick an instance type. Amazon EC2 offers many instance types tailored to specific use cases. **An instance is a virtual server or virtual machine.** They come in a variety of CPU, memory, storage, networking, and a lot more.



You can configure instance details, such as the number of instances, network, host type, and so on. Here, we'll use one instance and keep the remaining details default.

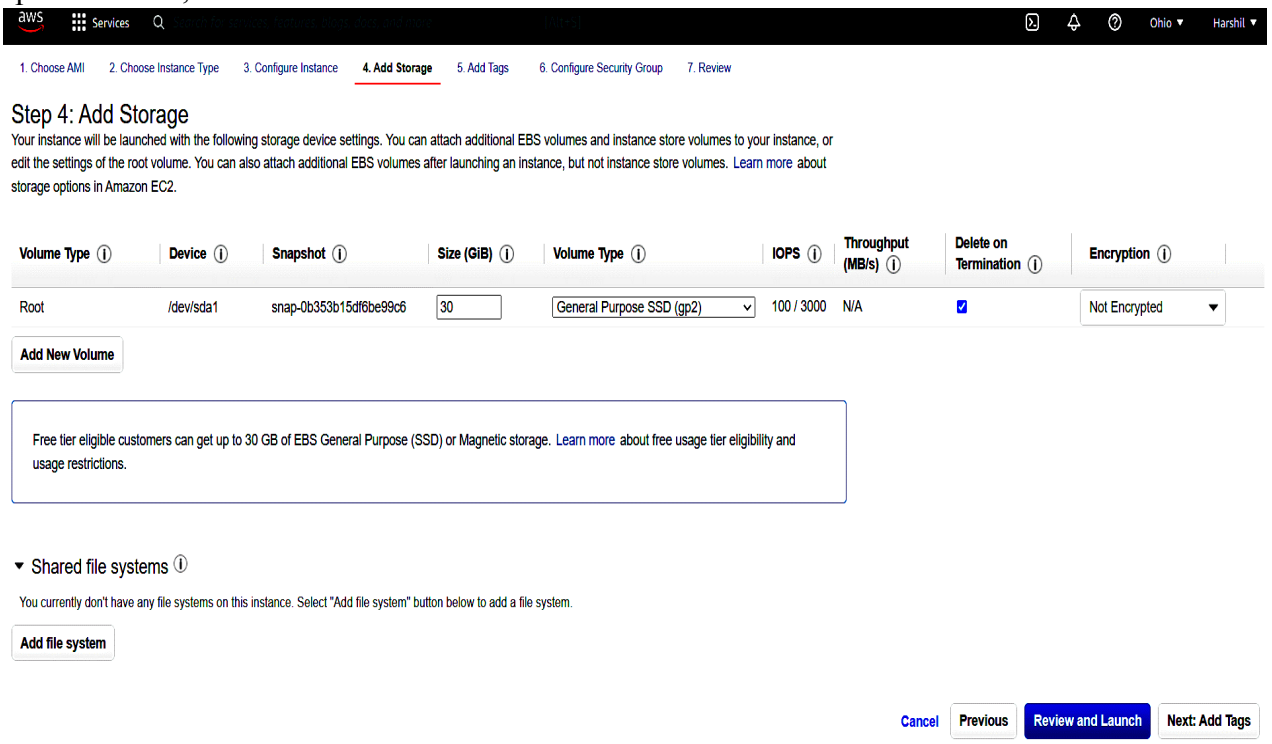


5. Add storage and tags

Once you configure an instance type, you can add or update storage info. AWS allows you to add more EBS volumes and instance store volumes, as well as change the root volume's parameters. *Amazon*

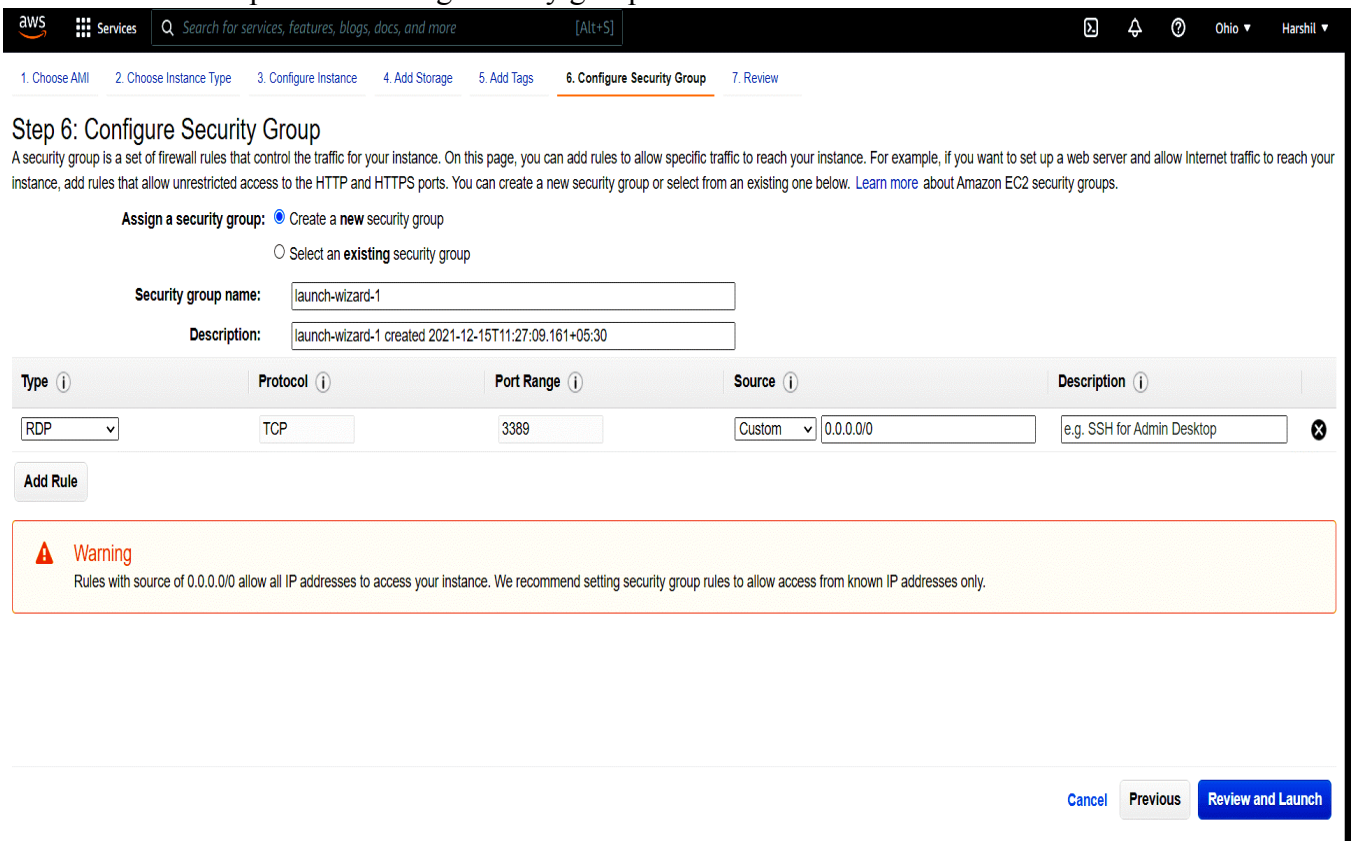
Elastic Block Store (EBS) provides block-level storage volumes for use with EC2 instances. It behaves like raw, unformatted block devices. You can mount these volumes as devices on your instances.

The next step is adding tags. A **tag is a label** applied to an AWS resource. Each tag has a key and an optional value, which users define.



6. Configure security

A **security group** is a set of firewall rules that control data entering and exiting your instance. You may either recreate it or pick an existing security group.

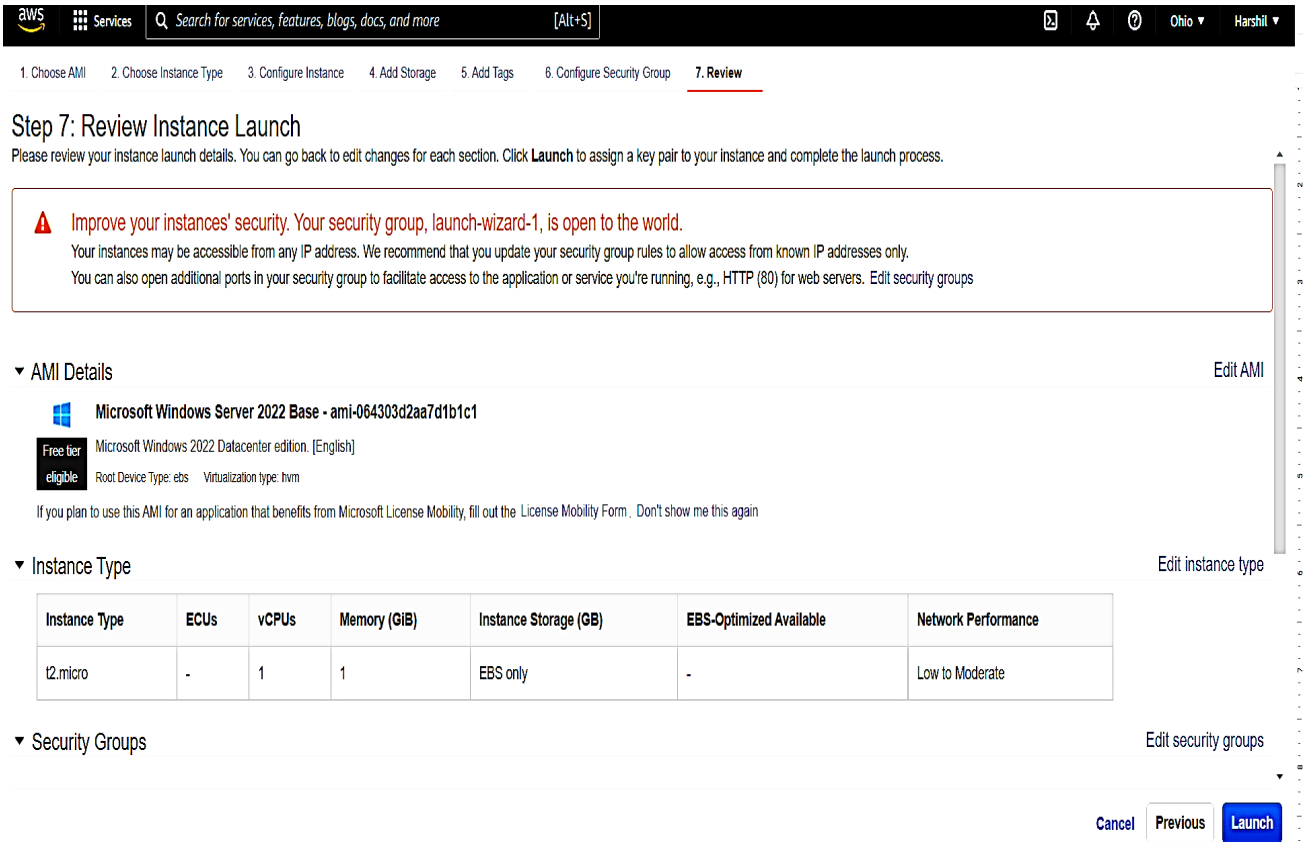


Security is a major concern when working on public clouds like AWS and Google Cloud Platform (GCP). Attackers can launch different attacks on public cloud deployments for lack of provider security. These attacks include DOS, DDOS, website defacement, and brute-force.

- Public clouds have **poor security**, but with the right set of rules, they can be improved.
- Public clouds offer **limited customization**. Clients can choose the operating system and size of the virtual machine.
- Cloud **data breaches** are frequently caused by misconfigured cloud security settings. Many companies' cloud security posture management solutions are insufficient for safeguarding their cloud-based infrastructure.

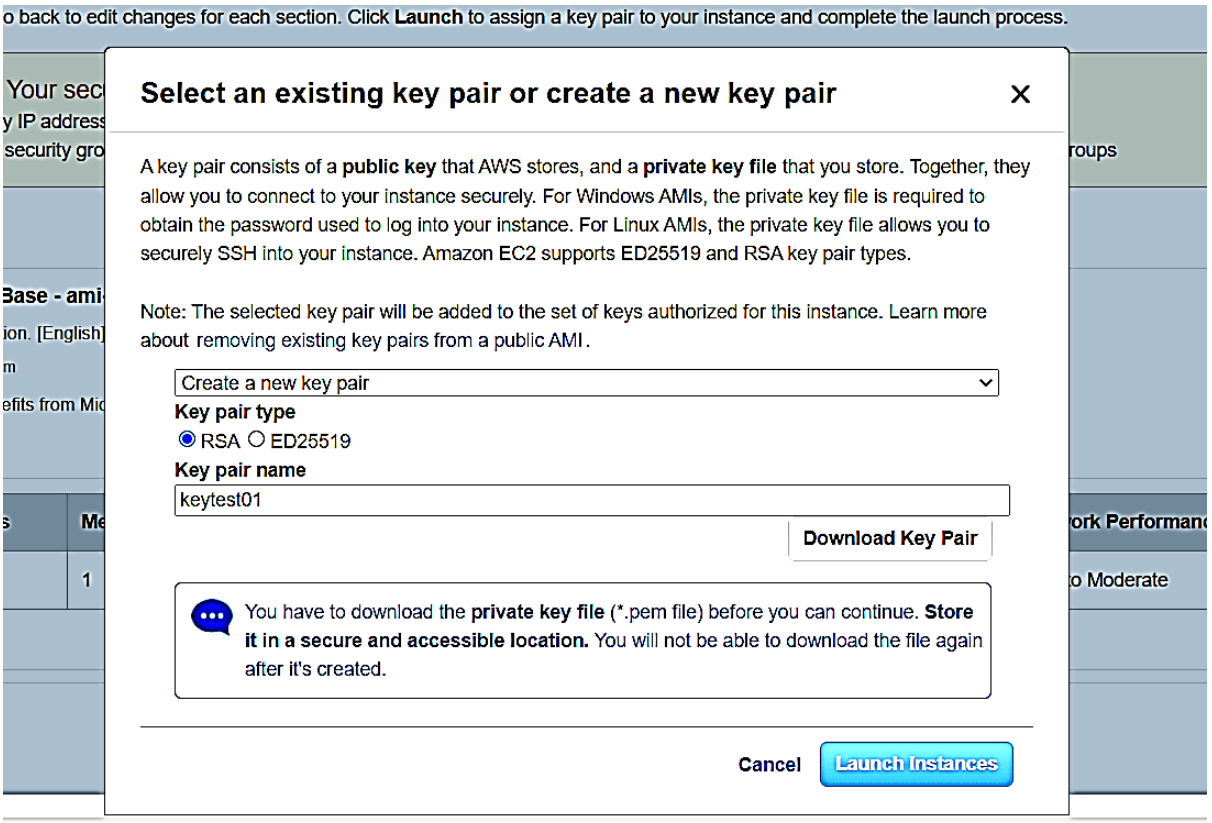
7. Review and launch your AWS virtual machine

The final step in creating an AWS virtual machine is to go through your instance details. Make sure every detail is correct, then click **"Launch"**.



When you click "Launch," you need to provide a key. To create a new key, select **"Create a new key pair"** from the drop-down menu and set a key name, for example, keytask, keytest1, and so on. Make sure you **download "key pair" before** launching your instance**.**

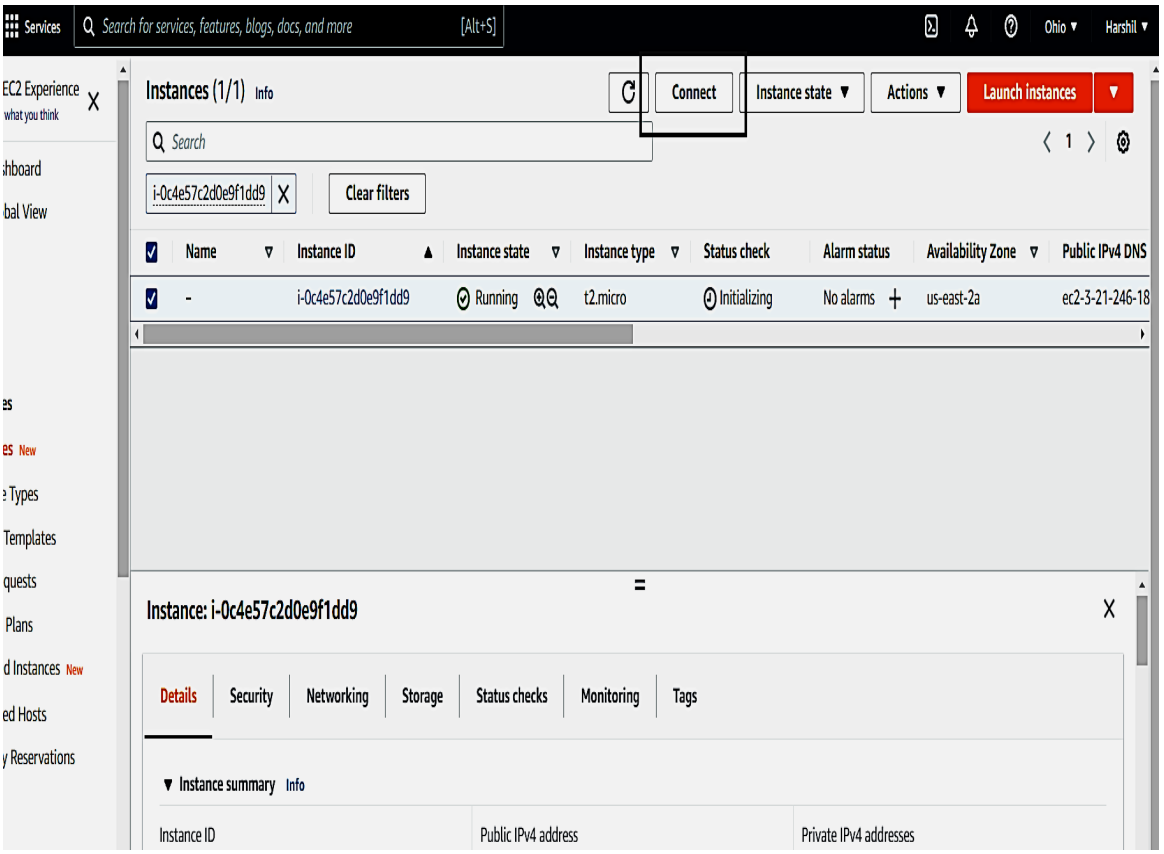
A key pair is made up of a public key stored by AWS and your private key file. They work together to allow you to connect to your instance safely.



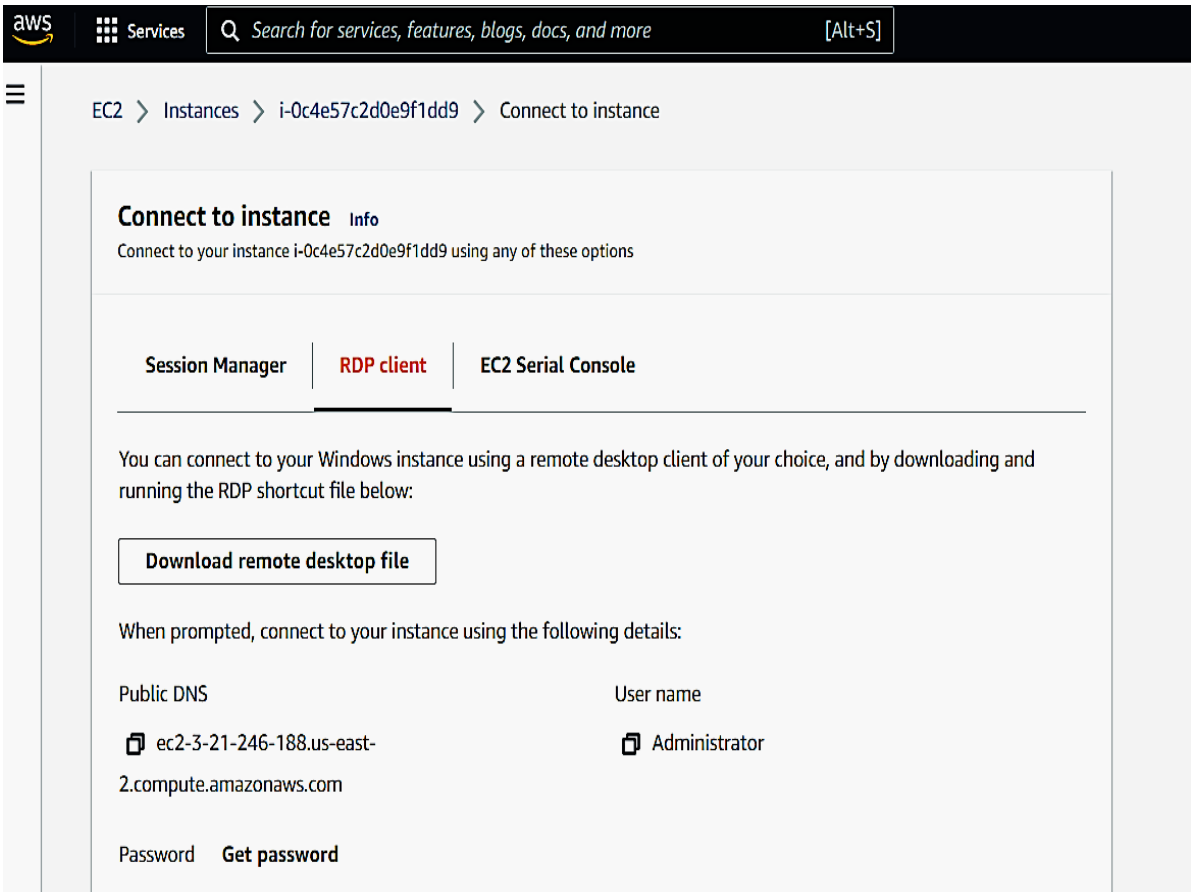
Voila! You successfully created and launched a virtual machine on AWS. Now, check the launch status and connect to an instance.

8. Connect to an instance

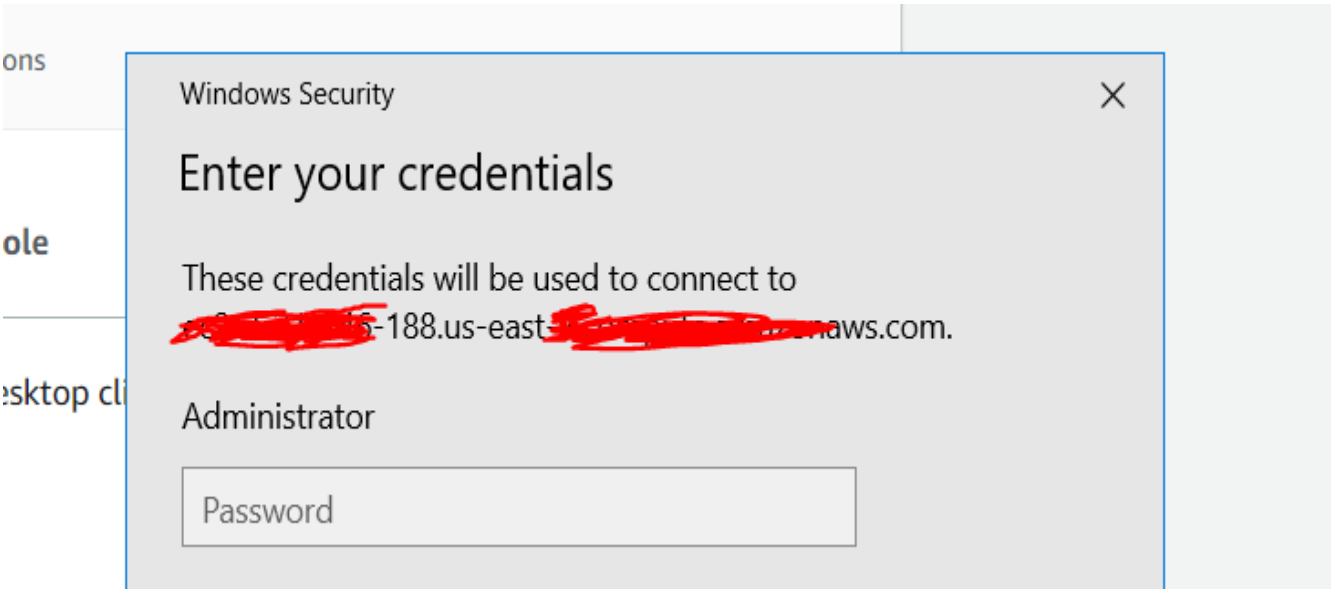
After starting the instance, you can check the status using **"Dashboard>Instances"**. Select your instance in the instance dashboard and click **"Connect"**.



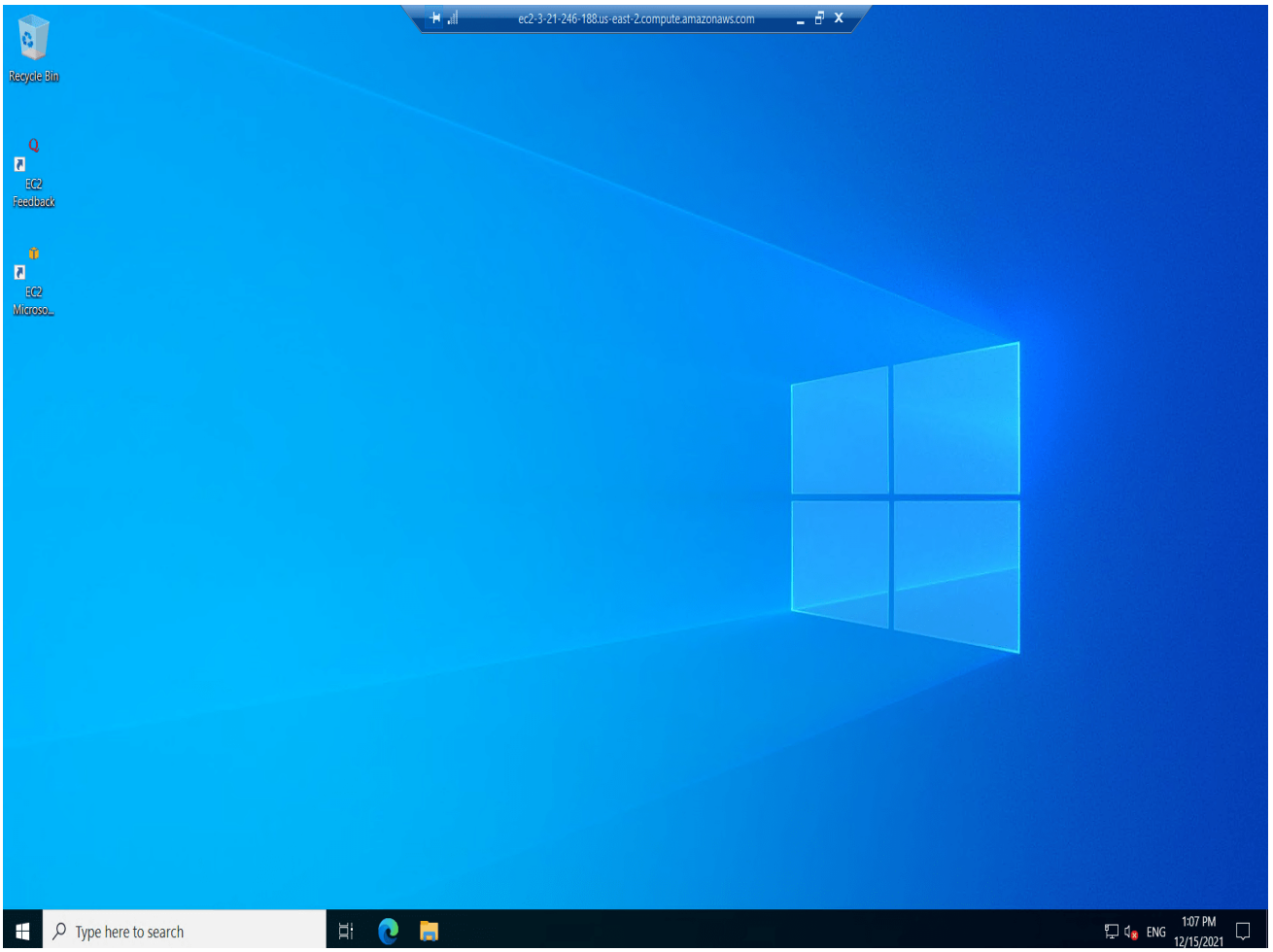
Select "**RDP client,**" click "**Get password,**" then upload the key pair downloaded when the instance launched (in step 7). After uploading the file, click **"decrypt password"** and download the remote desktop file**.**



Open the downloaded file and enter your password.



You should now see a screen similar to the one below, indicating that your **AWS Windows virtual machine** successfully launched!



Optimize your resources with a virtual machine

Software developers are always looking to increase software efficiency for better usage and management. As explained above, virtual machines can be a great asset to both businesses and individuals, helping them get rid of the hassle of physical server management. All they need is a deep understanding of what VMs are and how they're created.

Having an automated infrastructure to manage all of your applications is extremely beneficial when working on the cloud. Server outages and downtime are common when scaling infrastructure to meet the increasing demand.

Middleware can help you save your AWS cost by nearly 60% with its AI-powered autoscaling solution. Request early access today to gain a competitive advantage over others!

Createasimplewebappusingcloudservices

How to Create a Website for Free Using Amazon Web Services

Creating a website is very much tedious and costly in earlier days. Registering a domain, hosting, and configuring DNS settings were paid services through different service providers. Those days are thankfully over, as companies like Amazon now offer comprehensive functionality when it comes to creating and maintaining a website.

Table of Contents

Introduction

Amazon offers a complete set of tools for website management. In fact, most of these tools are available under Amazon's 12-month free tier, meaning that you can start a website from scratch with no money upfront! The only thing you'll have to pay for is a separate domain.

In this article, we're going to cover the following aspects of creating a website:

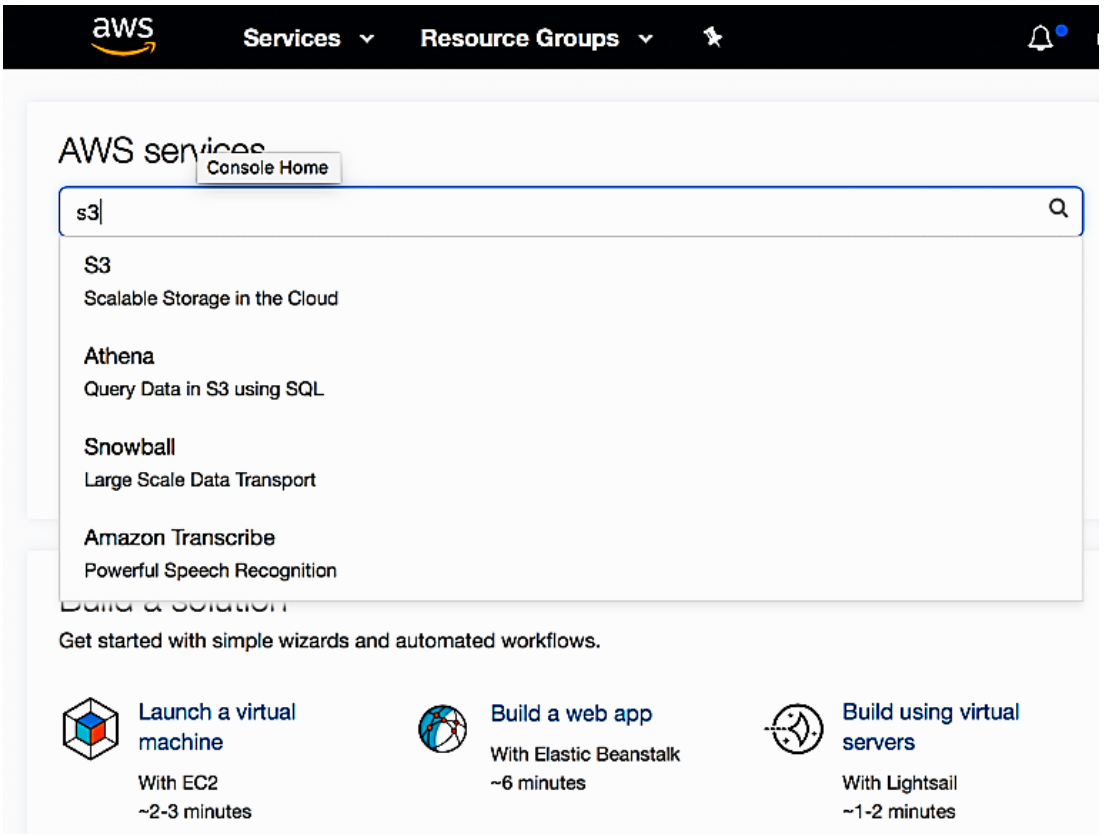
1. Hosting the website's assets using Amazon S3 (free);
2. Reducing latency and increasing transfer speeds by leveraging AWS CloudFront (free);
3. Registering a domain and redirecting it to CloudFront (extra fee for the domain).

How to Host a Website on Amazon S3

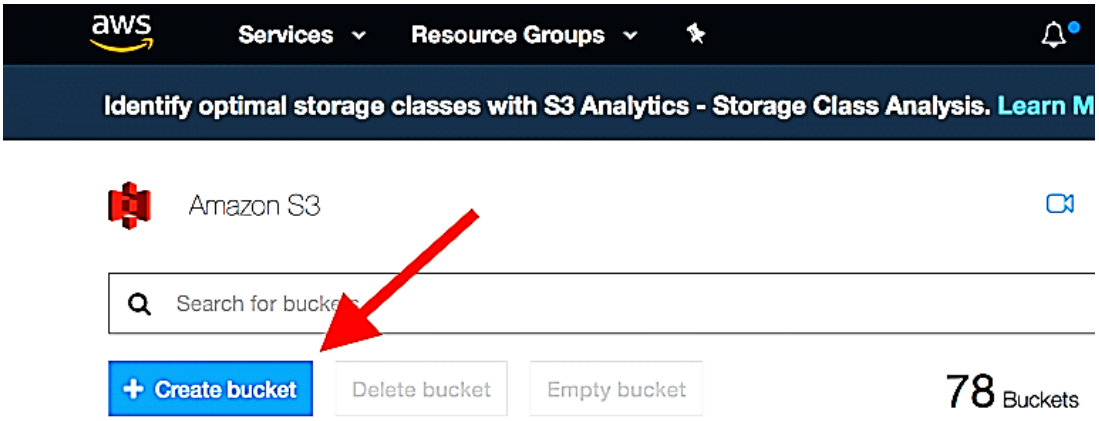
We need to arrange hosting for our website's assets (all of the HTML files, images, etc). For this purpose, Amazon offers their cloud object storage—Amazon S3. To get started with S3, you need to first sign up for the AWS Web Console.

Under the AWS Free Tier, you get 5 GB of standard storage, 20,000 Get Requests, and 2,000 Put Requests free of charge.

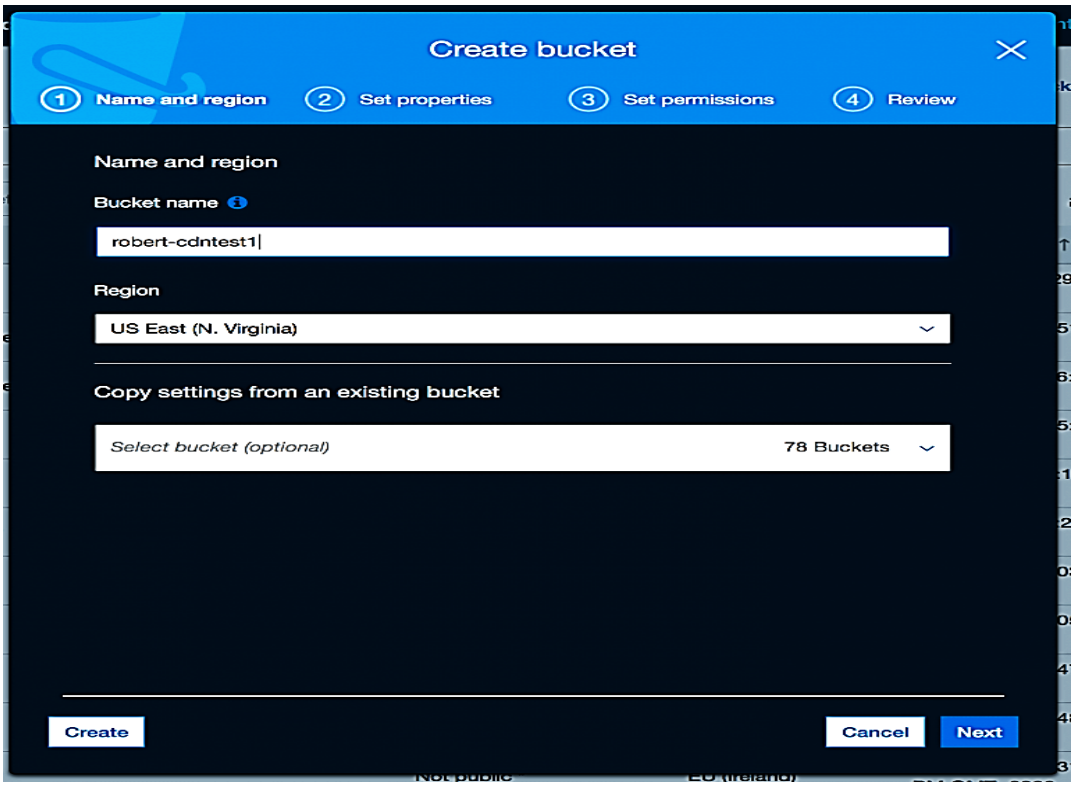
Once you've signed up for the console, open it, and search for S3, Amazon's object storage solution.



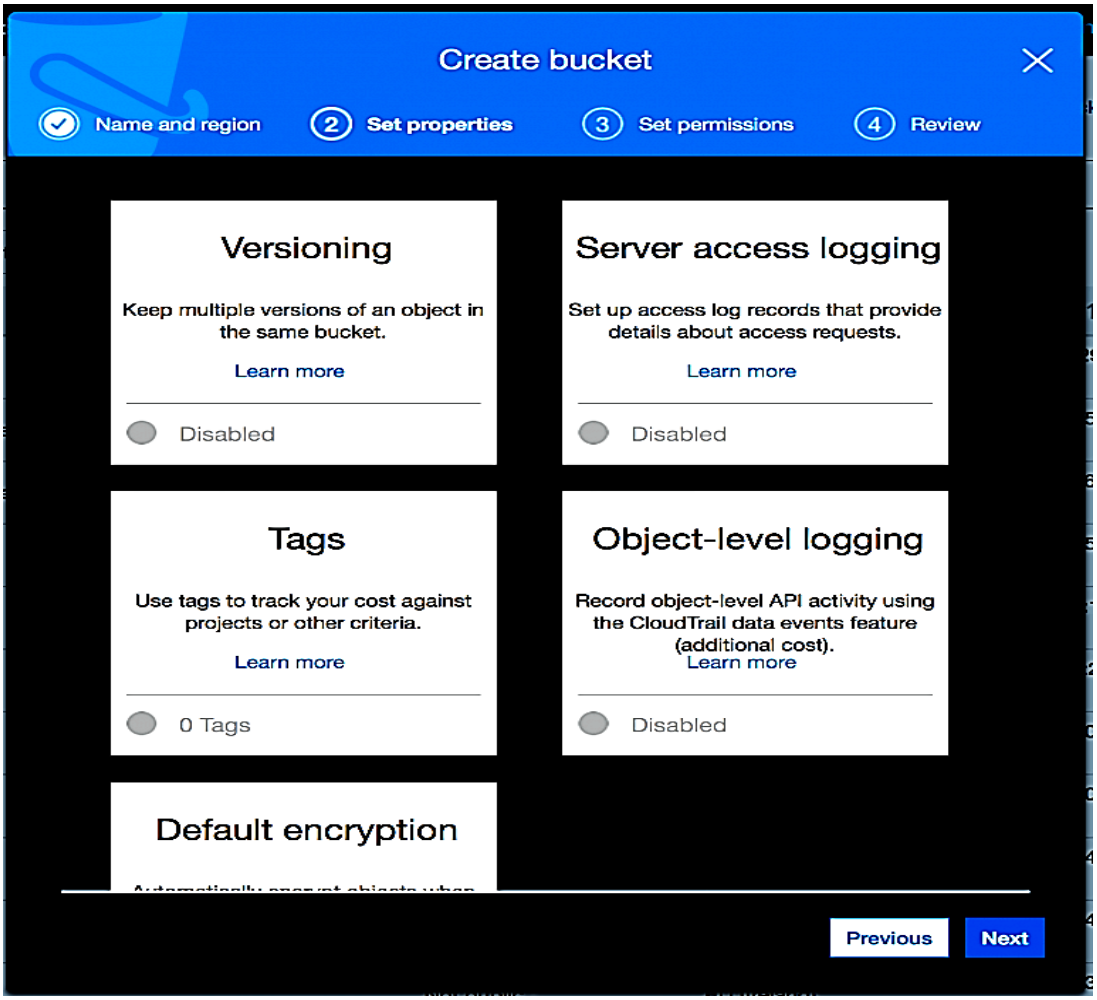
Now click **Create bucket**.



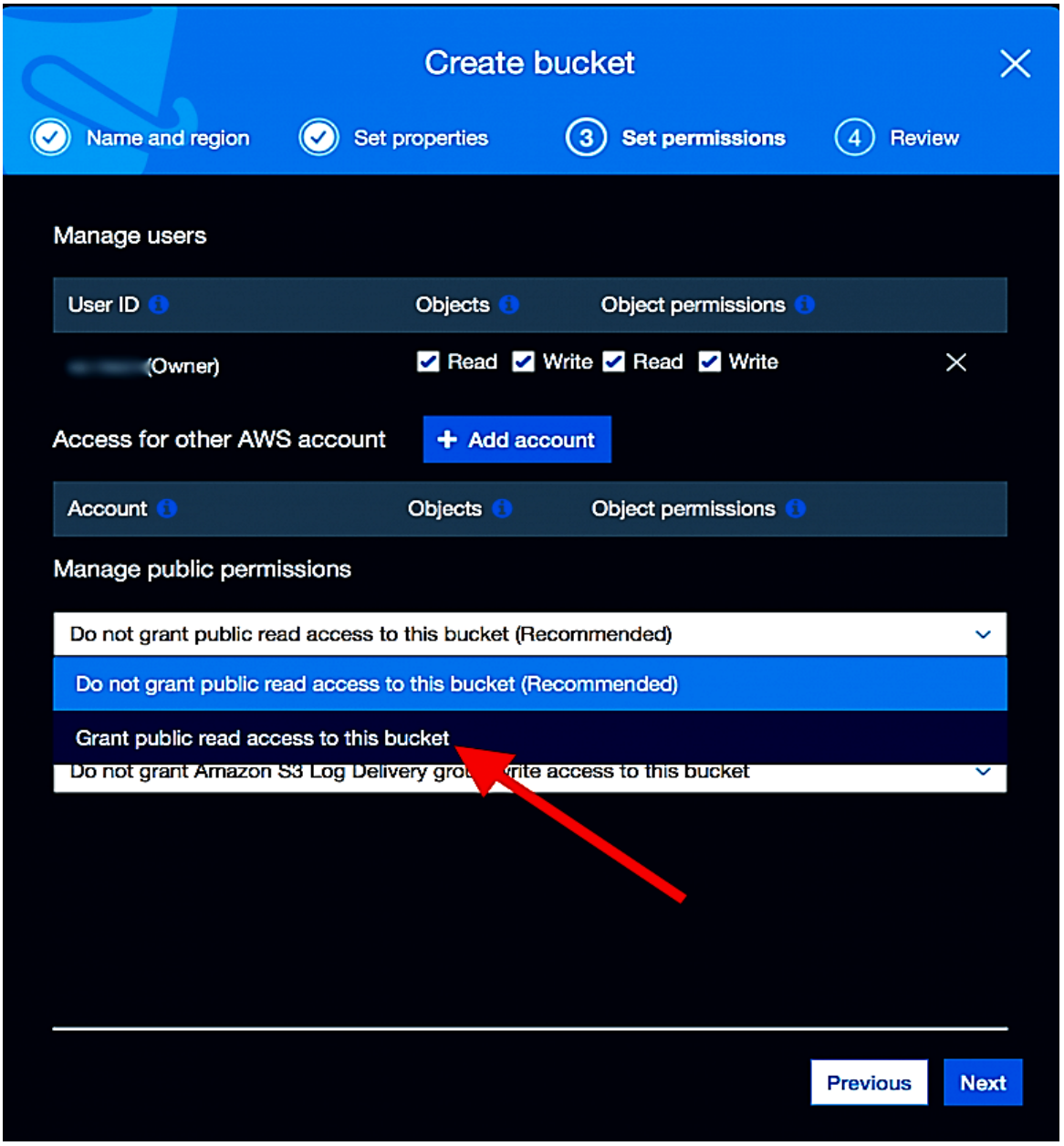
Enter the bucket's name, specify the region and click *Next*.



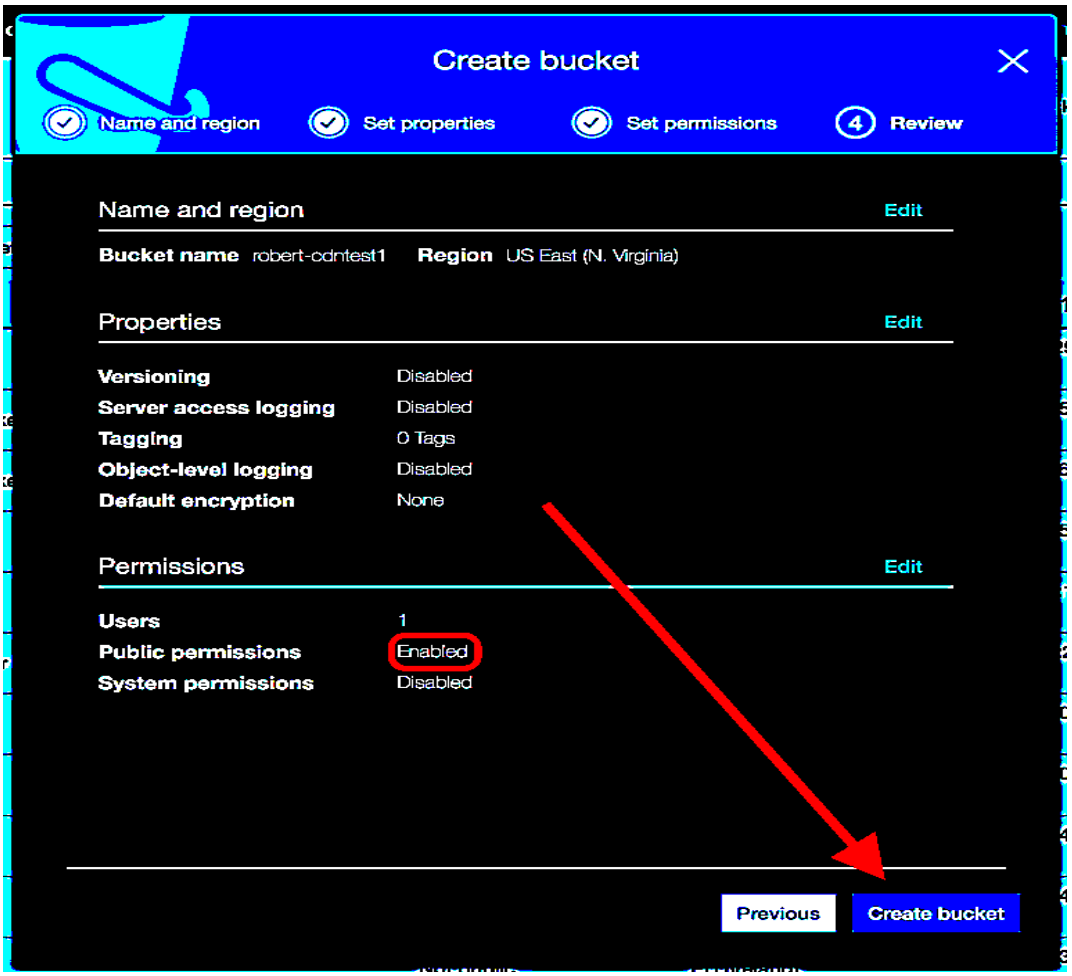
In the next step, you can enable versioning, server access logging, tagging, object-level logging, and default encryption. We don't need that for now, so just skip these options and click **Next**.



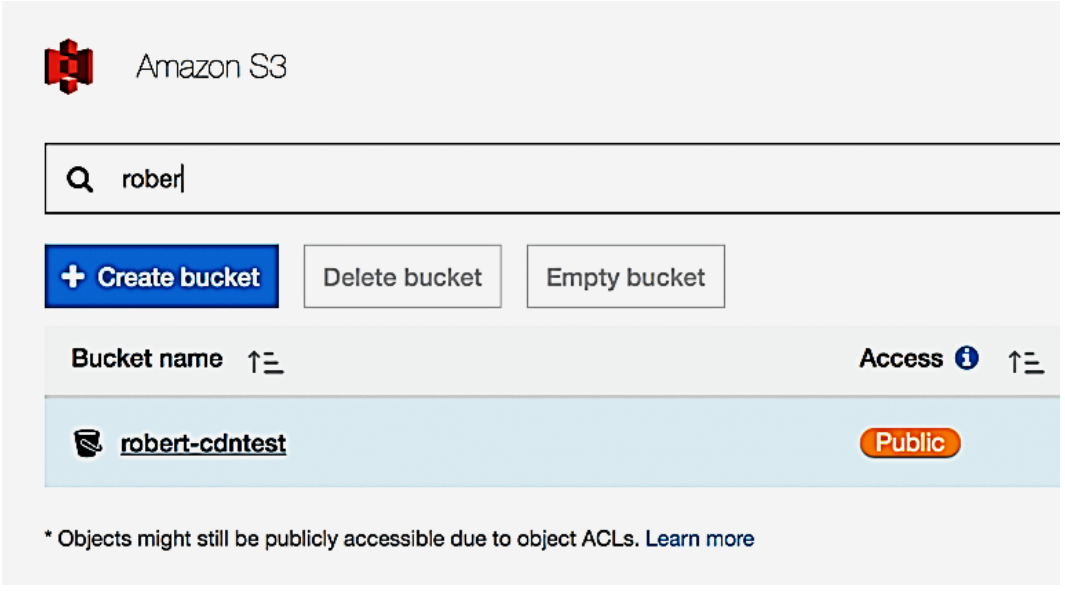
In the next step, you can specify the required permissions. It is important that you select **Grant public read access to this bucket** in the public permissions drop-down menu. Otherwise, it will be impossible to access the website's contents located in this bucket.



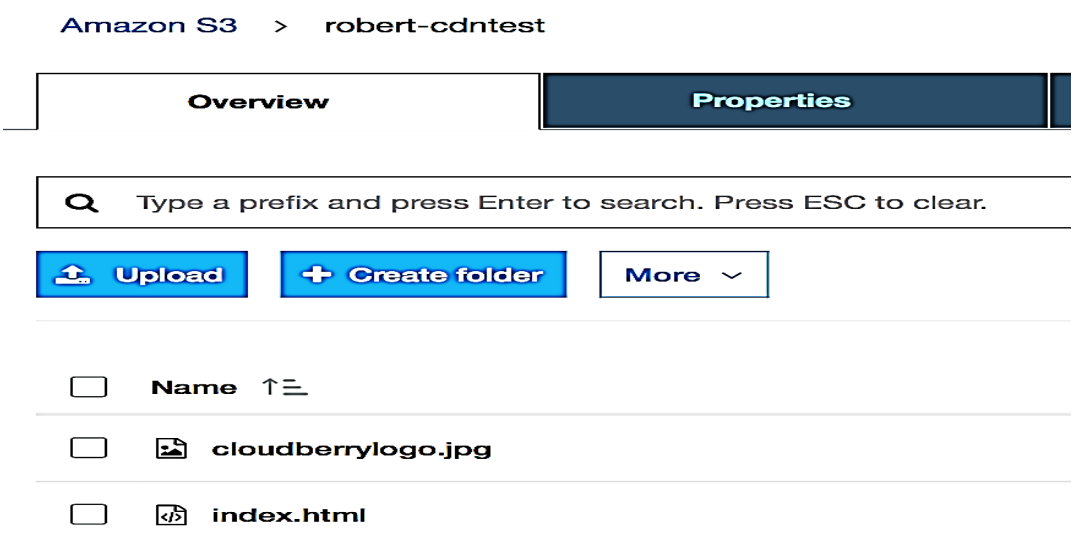
Review the bucket's properties and permissions, and click **Create bucket**.



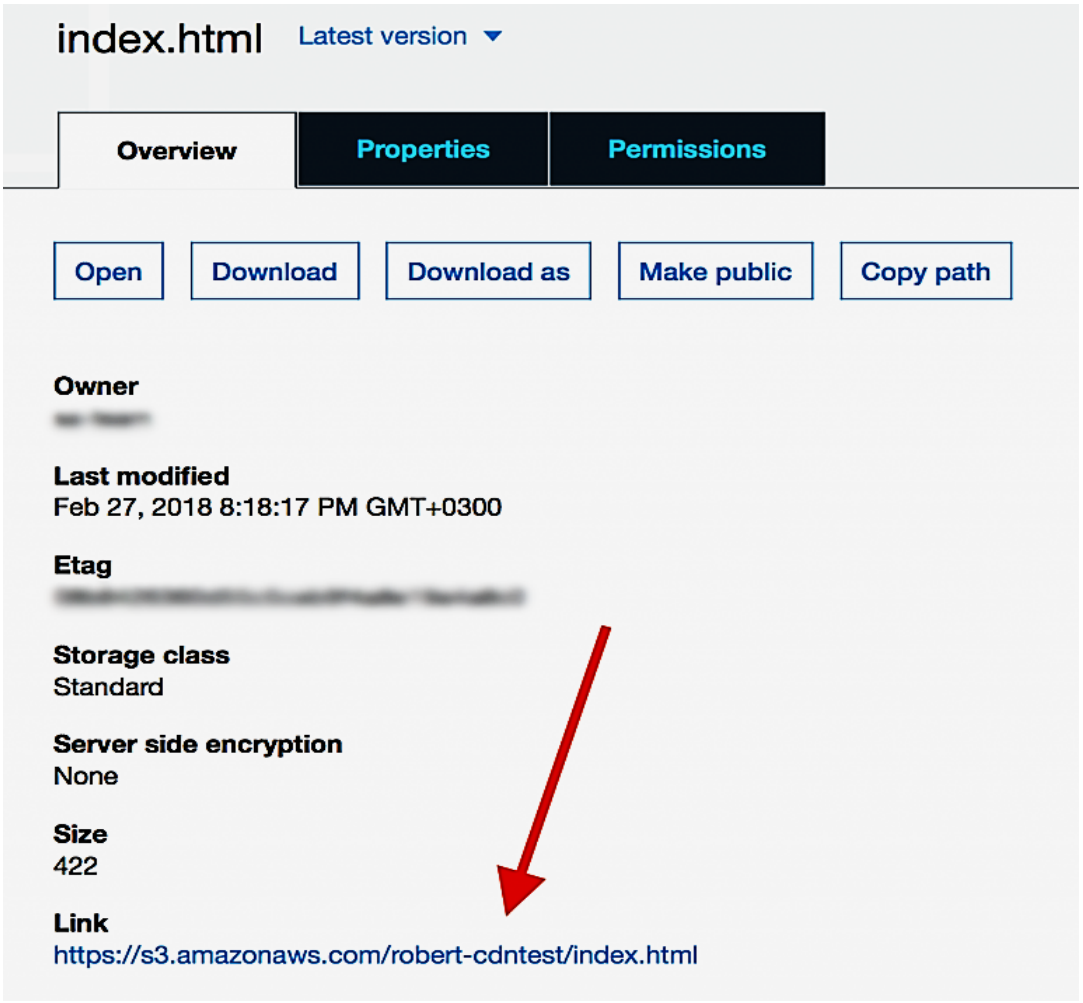
Now, locate your bucket in the bucket list and click on it.



Click **Upload** and upload the assets. In our case, we'll simply add an HTML file and an image to host a simple static page.



Our hosting is now ready, and we can access the assets from anywhere. Just click on the HTML file and try to open the link. You should see the webpage you've uploaded to the bucket. If you only want to use AWS for hosting and want to get the domain elsewhere, you're done! Register your domain at a different company and point it to the link displayed in the screenshot below.

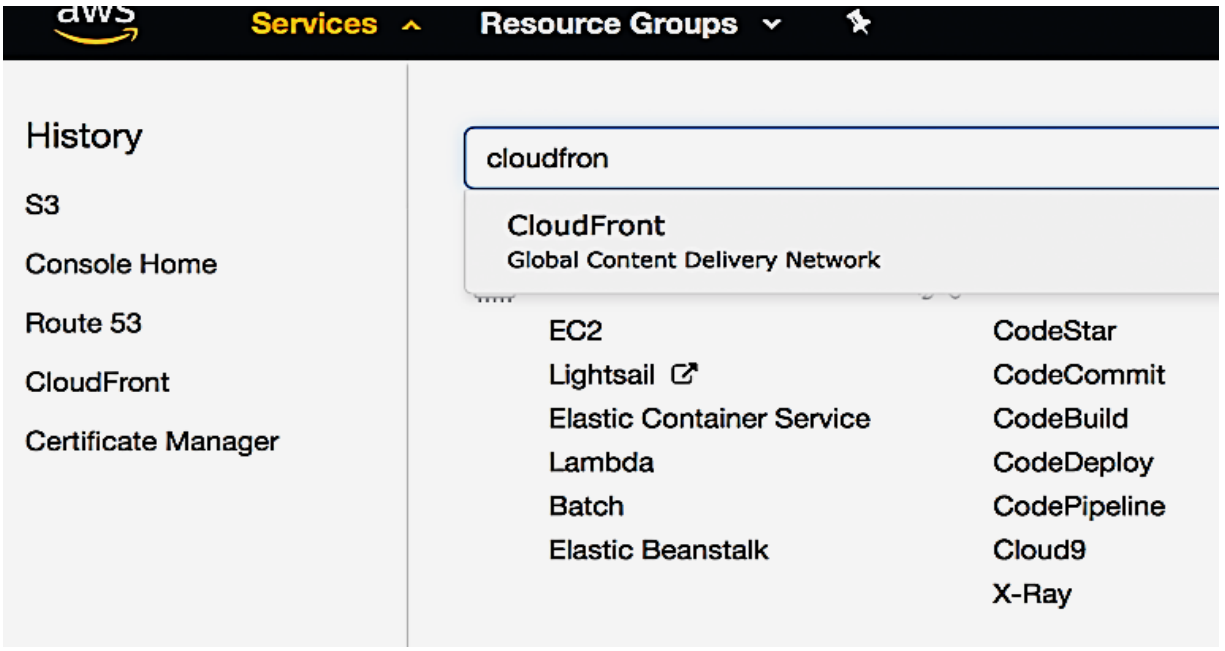


Now that we've covered hosting, let's move over to network optimization with CloudFront.

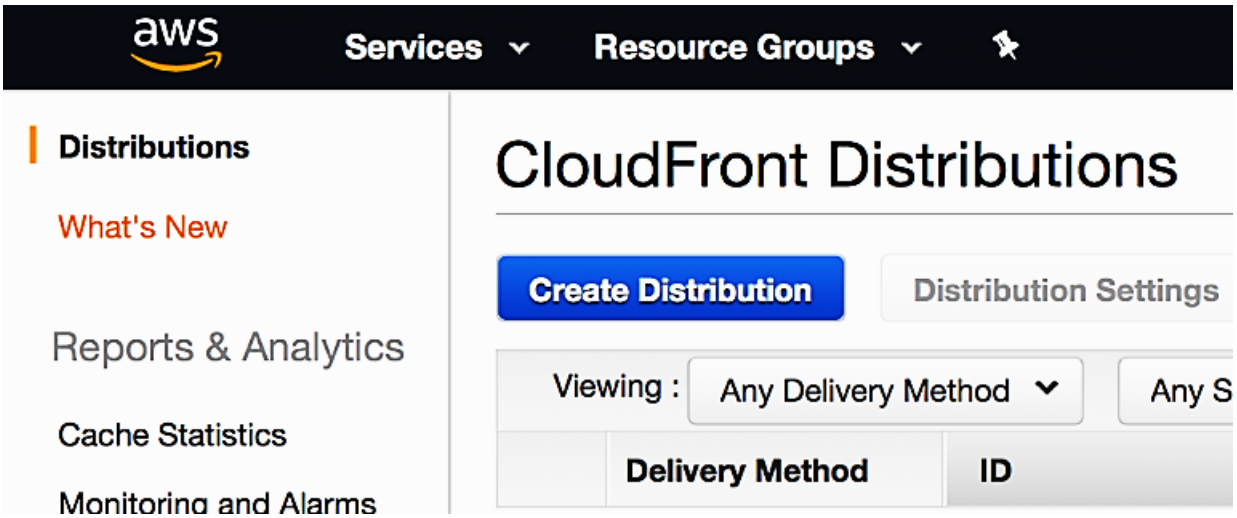
Leveraging Amazon CloudFront

Amazon CloudFront is a global content delivery network (CDN) service that securely delivers data to your users with low latency and high transfer speeds. CloudFront delivers your content through a worldwide network of data centers called edge locations. When a user requests content that you're serving with CloudFront, the user is routed to the edge location that provides the lowest latency (time delay), so that content is delivered with the best possible performance.

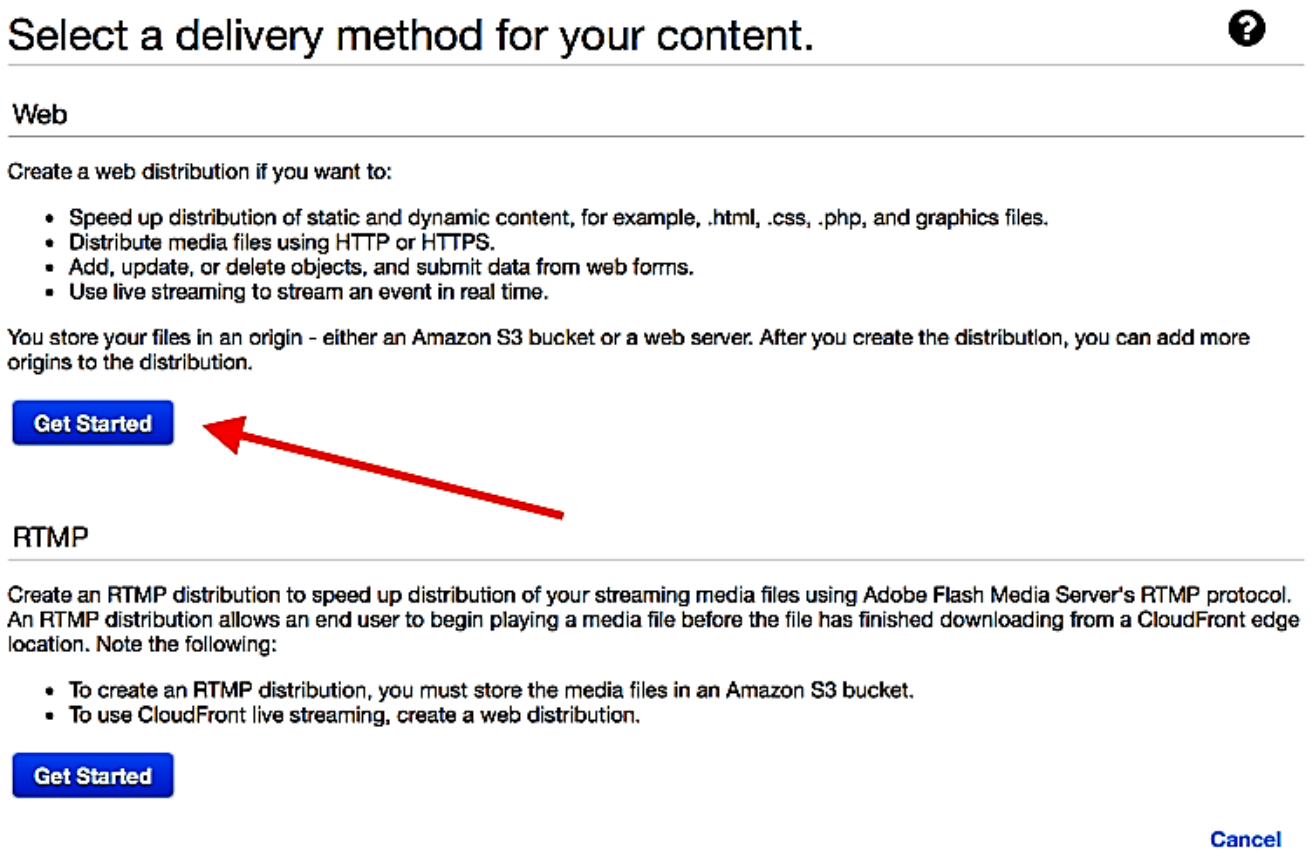
Search for CloudFront in the AWS Console search bar. Open it.



At this point, you need to create a CloudFront distribution to ensure that your website's assets are accessible from anywhere quickly. Click **Create Distribution**.



Click **Get Started** under *Web*.



The next step is one of the most complicated. Here, you need to specify the distribution's origin settings, default cache behavior settings, and general distribution settings. This lengthy article from Amazon provides an in-depth description of each option.
We'd like to point out just a few essential options that you need to configure:

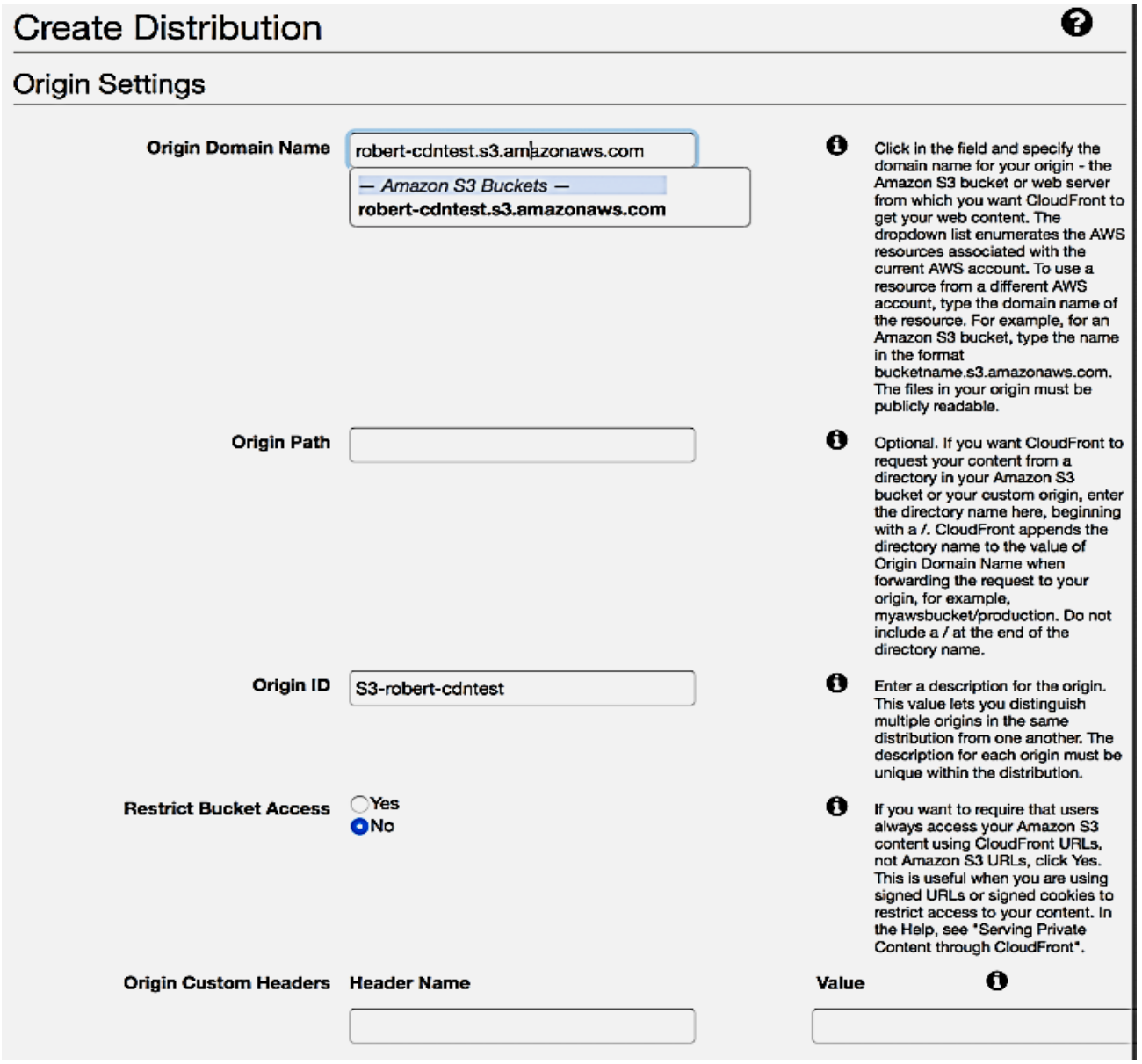
Origin Domain Name. This is the source bucket that contains all the data. Once you start typing in the bucket's name, Amazon will automatically suggest the right option.

Alternate Domain Names (CNAMEs). If you want to use subdomains to access your website—as we're going to do—specify them. We're going to enter *test.cloudberrylab.io.*

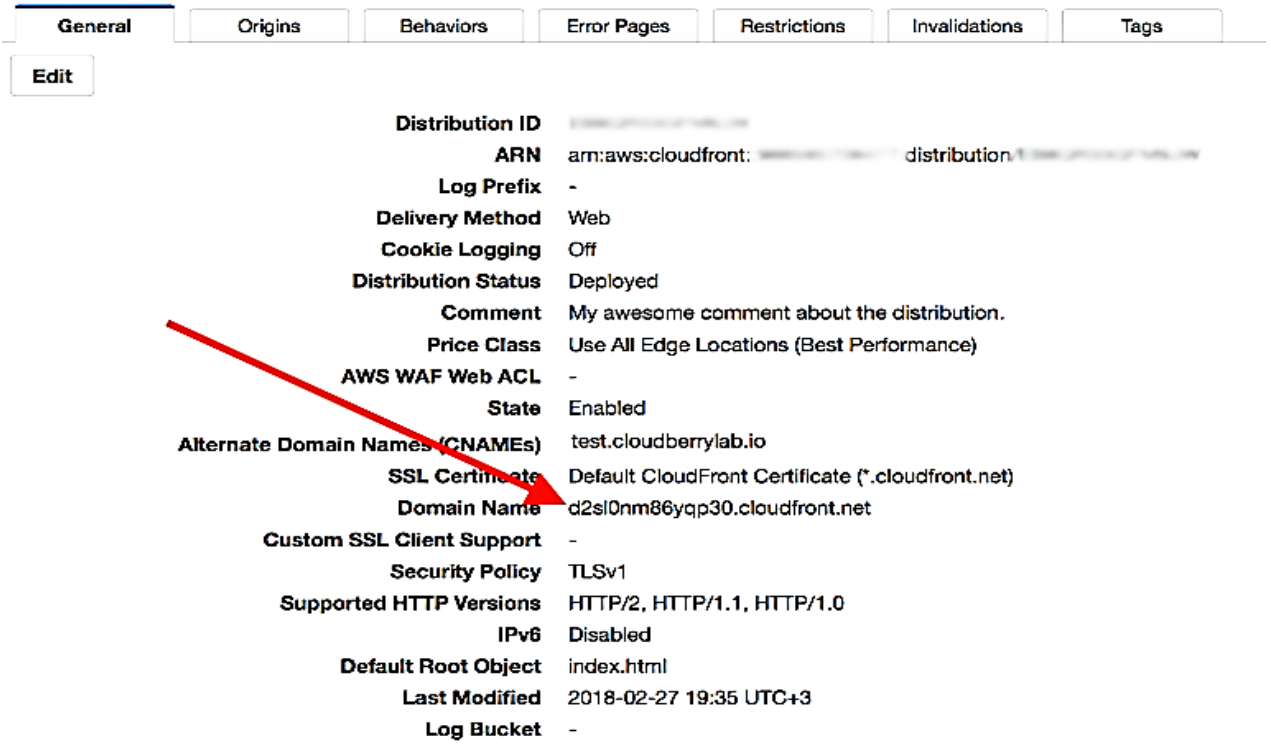
Default Root Object. By default, if you try to open the website's address, you'll encounter an error because *test.cloudberrylab.io* is not an HTML page, while *test.cloudberrylab.io/index.html* is. At this point, requiring users to specify the index page is outdated. To enter the page's name, and users will automatically be redirected

from*test.cloudberrylab.io to test.cloudberrylab.io/index.html.* Do not write "/index.html"—simply enter "index.html" without a slash.

IPv6. It's easier if you disable IPv6 for now. If it's enabled, you'll have to create two alias records when redirecting your domain to CloudFront. You can do that later; for now, just disable it.



When done, click **Create Distribution**. Once it's created, click on it, and the console will display the general information. The **Domain Name** is your CloudFront address. When accessing the website's content using this link, users will receive the data from the fastest location. Copy and paste it into the browser address field and hit *Enter*. You should see your page!



It's important that your HTML files refer to the images using the CloudFront addresses to ensure that images are loaded quickly. Every asset is available at the following address:

CloudFront Domain name + the asset's path in the S3 bucket.

Here's how that looks within our sample HTML file:

```
<html>

<body>

    <p>This static page is stored in an S3 bucket and can be quickly reached with the help of
    CloudFront. The image is loaded from the same S3 bucket. CloudFront and S3 are leveraged under
    the free 12-month tier while the domain name is provided for an extra fee.</p>


    <p><img style="max-width:150px;" src="https://d2sl0nm86yqp30.cloudfront.net/cloudberrylogo.jpg">

</body>

</html>
```

And here's our page:

All of the assets are stored in S3 and are reached through CloudFront for the best performance.

Under the AWS Free Tier, you get 50 GB Data Transfer Out, and 2,000,000 HTTP and HTTPS Requests from Amazon CloudFront free of charge for 12 months.

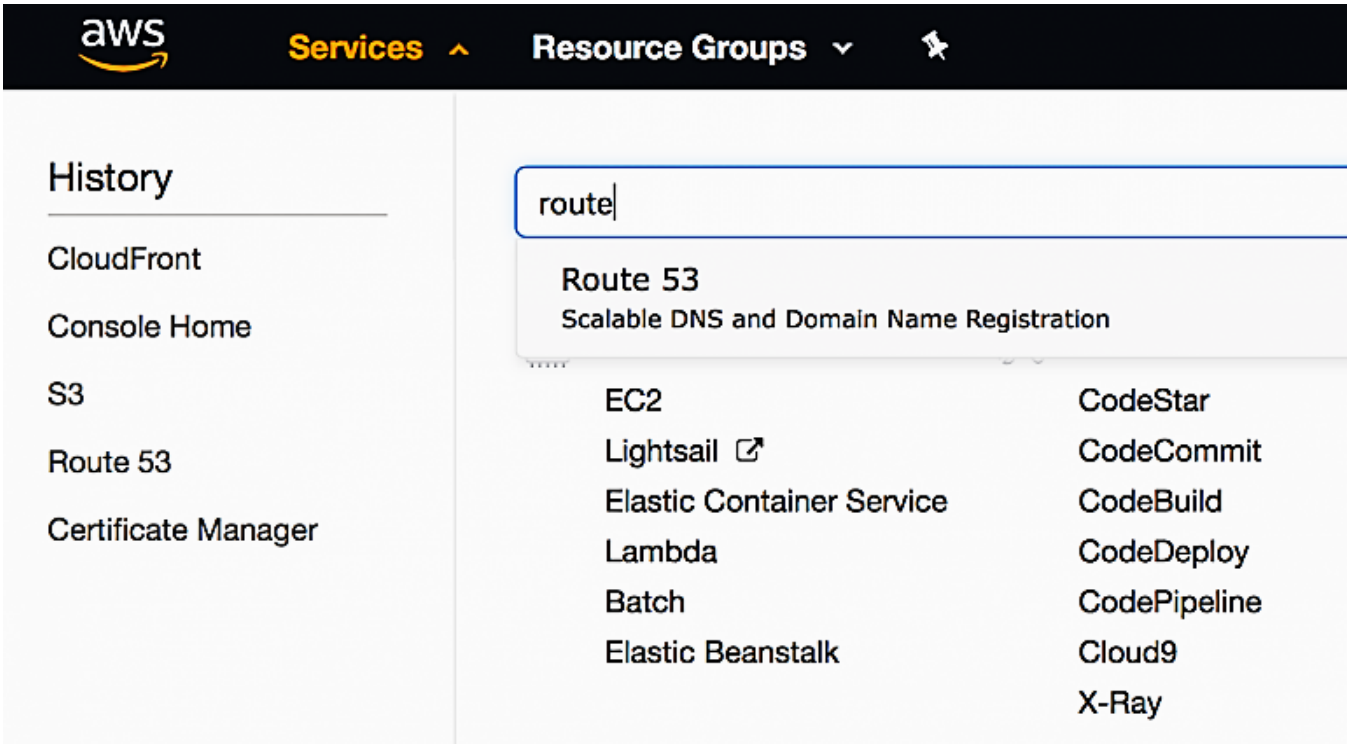
We're close to finished here. In the last section, we will explain how to register a domain and redirect it to CloudFront's domain name using AWS Route 53. Naturally, this comes at an extra price. So if you were looking only for the free solutions, you can close the article now and enjoy your website!

Registering a Domain Using AWS Route 53

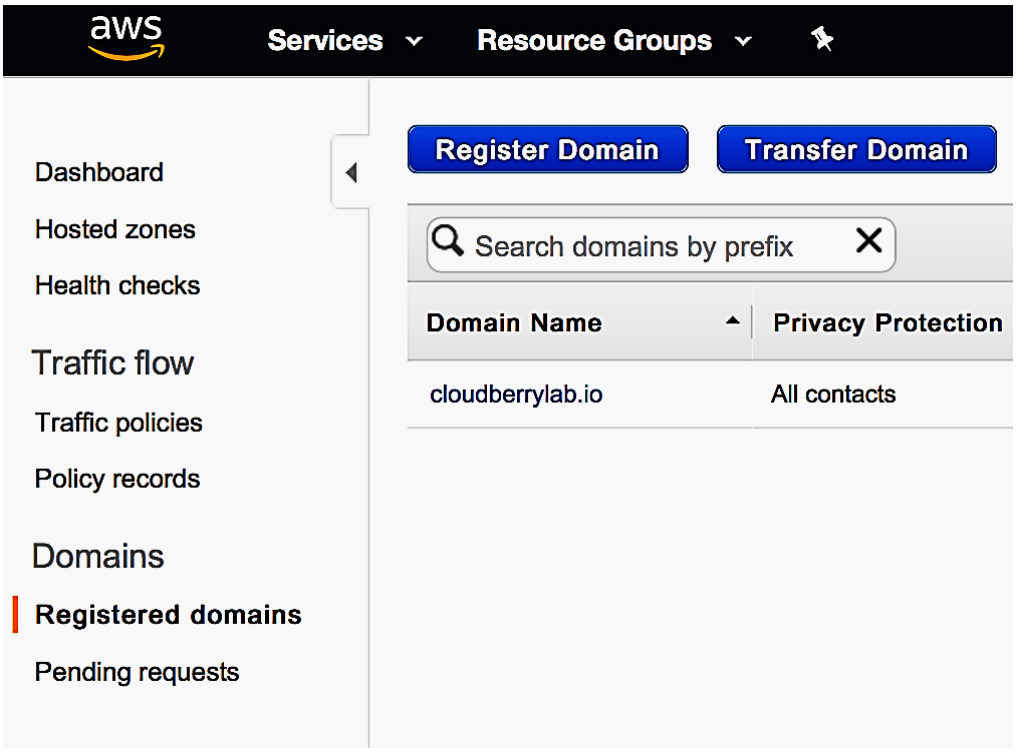
You can use Amazon Route 53 to help you get a website or web application up and running. Route 53 performs three main functions:

1. Registering domain names
2. Routing Internet traffic to the resources for your domain
3. Checking the health of your resources.

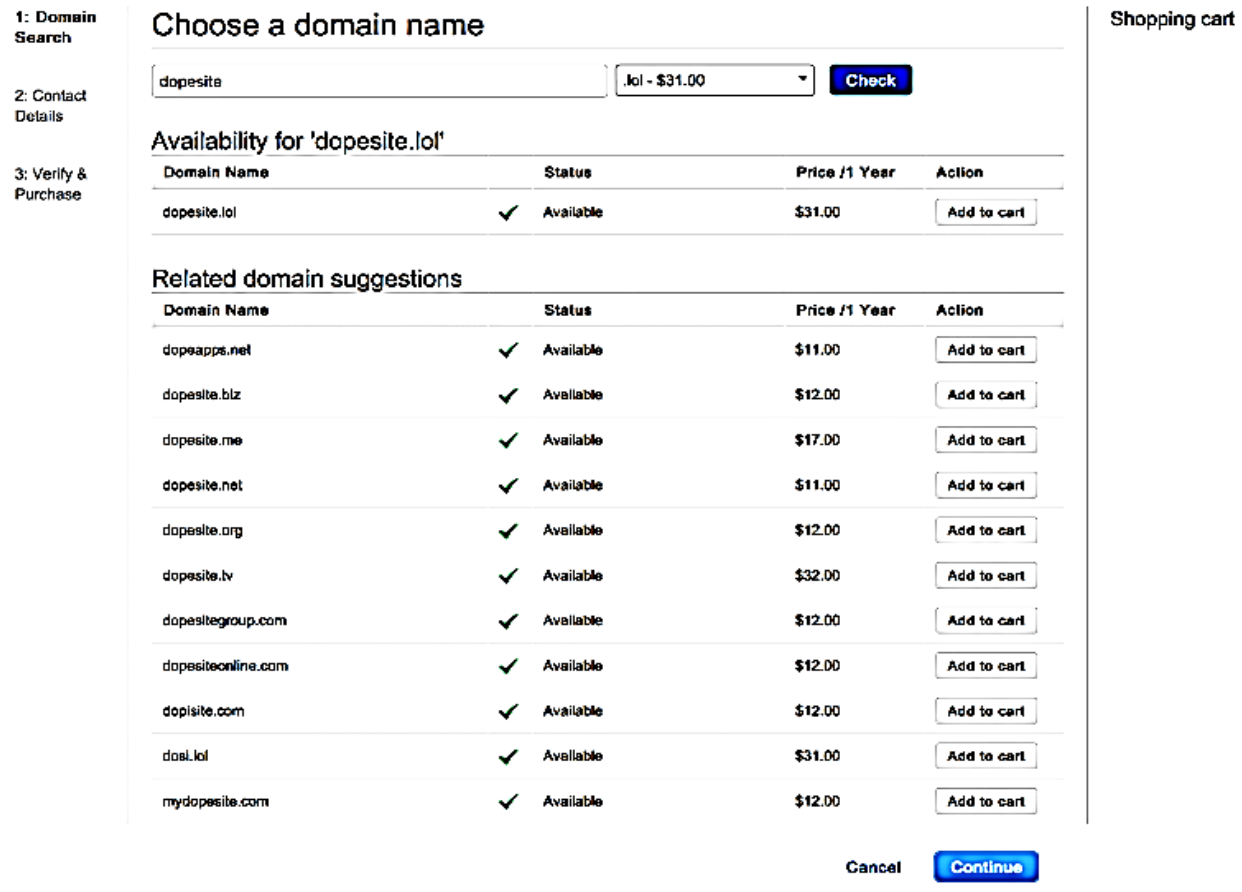
Go to the AWS Route 53 service.



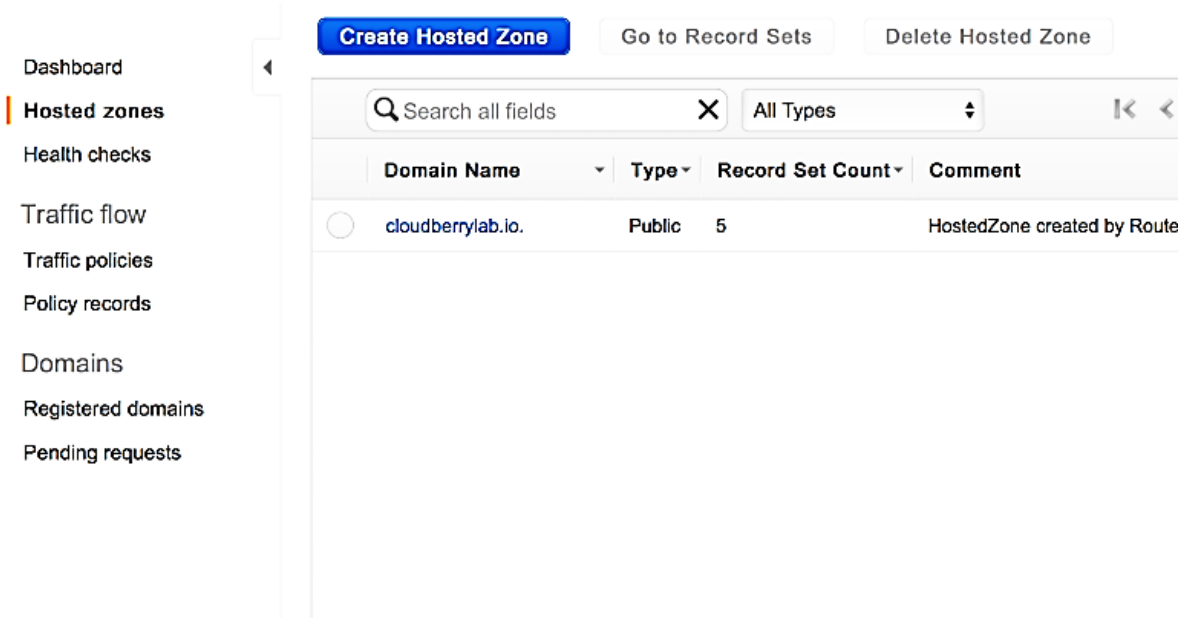
Under *Registered Domains*, click **Register Domain**.



Go through the three steps .



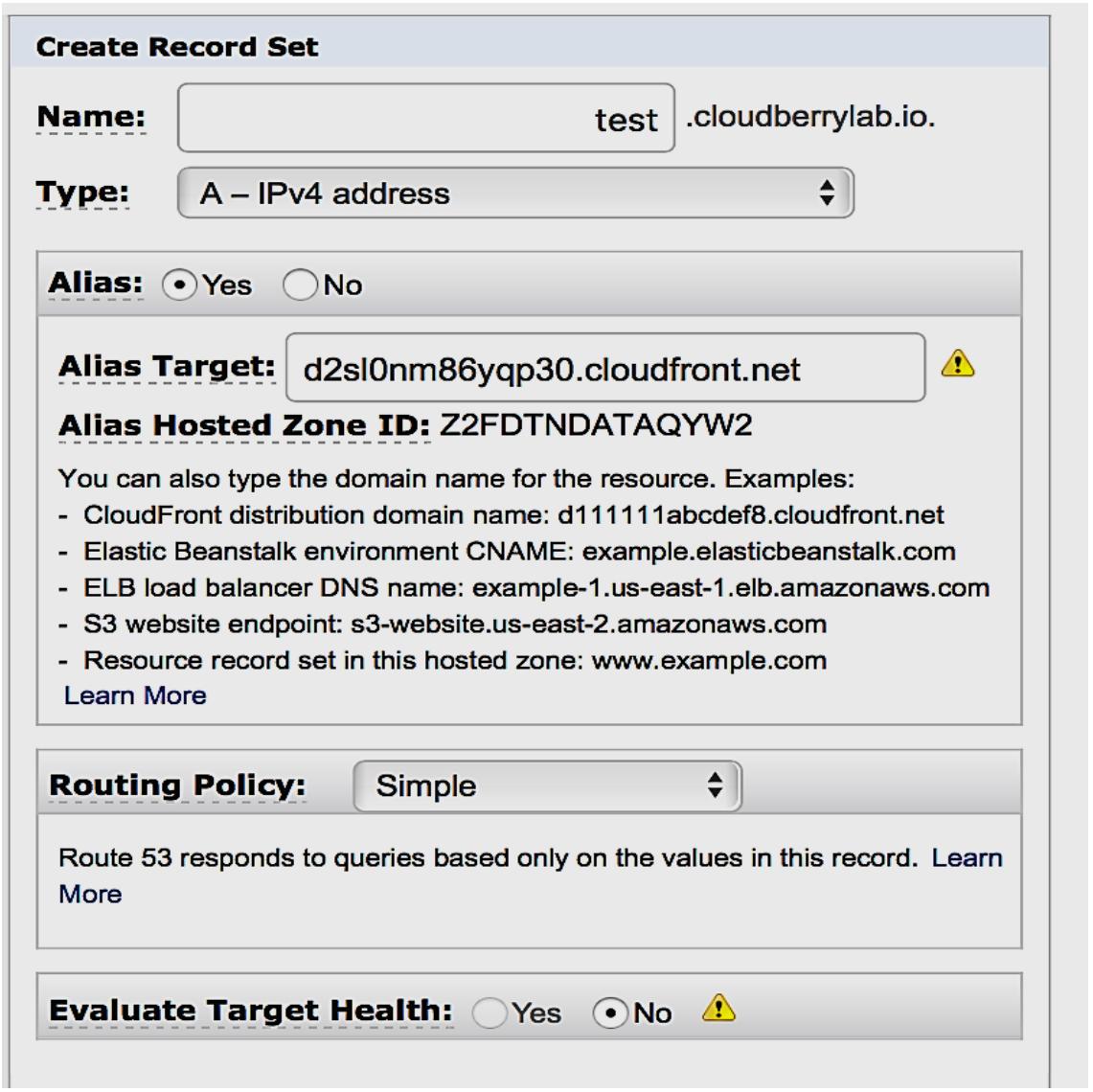
When done, go to the *Hosted zones* and click on your domain.



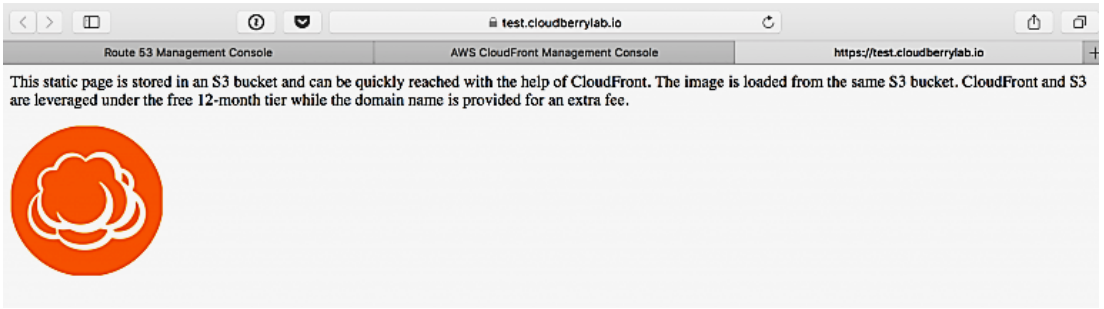
Now you need to create a r*ecord set* which will route Internet traffic to your CloudFront distribution. Click **Create Record Set**.
Now, specify the following parameters:

- **Name.** Enter the domain name that you want to use to route traffic to your CloudFront distribution. In this example we'll type in *test;*
- **Type**. Select **A – IPv4 address;**
- **Alias**. Select **Yes;**
- **Alias Target.** Enter the domain name of your CloudFront distribution;
- **Routing Policy.** Leave the default value of **Simple**;
- **Evaluate Target Health**. Select **No**.

Click **Create**.



t may take a couple of minutes for your traffic to be routed to your CloudFront distribution. If you did everything correctly, your content should be available at the specified domain.



Conclusion : Creating and maintaining a website using AWS is indeed a very easy task. Amazon's well-integrated ecosystem enables you to effortlessly register a domain, host your website, and ensure fast load times with the help of CloudFront.

W 4 D 1

Refer separate file named “htmlcssjscriptbasics.doc”

JSON

What is JSON?

- JSON stands for **J**ava**S**cript **O**bject **N**otation
- JSON is a lightweight data-interchange format
- JSON is plain text written in JavaScript object notation
- JSON is used to send data between computers
- JSON is language independent *

The JSON syntax is derived from JavaScript object notation, but the JSON format is text only.
Code for reading and generating JSON exists in many programming languages.
The JSON format was originally specified by Douglas Crockford.

JSON Example

This example is a JSON string:
'{"name":"John", "age":30, "car":null}'
It defines an object with 3 properties:

- name
- age
- car

Each property has a value.
If you parse the JSON string with a JavaScript program, you can access the data as an object:
let personName = obj.name;
let personAge = obj.age;

Why Use JSON?

The JSON format is syntactically similar to the code for creating JavaScript objects. Because of this, a JavaScript program can easily convert JSON data into JavaScript objects.
Since the format is text only, JSON data can easily be sent between computers, and used by any programming language.
JavaScript has a built in function for converting JSON strings into JavaScript objects:
JSON.parse()
JavaScript also has a built in function for converting an object into a JSON string:
JSON.stringify()
You can receive pure text from a server and use it as a JavaScript object.
You can send a JavaScript object to a server in pure text format.
You can work with data as JavaScript objects, with no complicated parsing and translations.

Storing Data

When storing data, the data has to be a certain format, and regardless of where you choose to store it, *text* is always one of the legal formats.
JSON makes it possible to store JavaScript objects as text.

JSON Syntax

The JSON syntax is a subset of the JavaScript syntax.

JSON Syntax Rules

- JSON syntax is derived from JavaScript object notation syntax:
- Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects
- Square brackets hold arrays

JSON Data - A Name and a Value

JSON data is written as name/value pairs (aka key/value pairs).
A name/value pair consists of a field name (in double quotes), followed by a colon, followed by a value:

Example
"name":"John"
JSON names require double quotes.

JSON - Evaluates to JavaScript Objects
The JSON format is almost identical to JavaScript objects.
In JSON, *keys* must be strings, written with double quotes:

JSON
{"name":"John"}
In JavaScript, keys can be strings, numbers, or identifier names:
JavaScript
{name:"John"}

JSON Values
In **JSON**, *values* must be one of the following data types:
- a string
- a number
- an object
- an array
- a boolean
- null

In **JavaScript** values can be all of the above, plus any other valid JavaScript expression, including:
- a function
- a date
- undefined

In JSON, *string values* must be written with double quotes:
JSON
{"name":"John"}
In JavaScript, you can write string values with double *or* single quotes:
JavaScript
{name:'John'}

JSON Files
The file type for JSON files is ".json"
The MIME type for JSON text is "application/json"

JSON Data Types

In JSON, values must be one of the following data types:
- a string
- a number
- an object (JSON object)
- an array
- a boolean
- *null*

JSON values **cannot** be one of the following data types:
- a function
- a date
- *undefined*

JSON Strings
Strings in JSON must be written in double quotes.
Example
{"name":"John"}

JSON Numbers
Numbers in JSON must be an integer or a floating point.

Example
{"age":30}

JSON Objects
Values in JSON can be objects.
Example
{
"employee":{"name":"John", "age":30, "city":"New York"}
}
Objects as values in JSON must follow the JSON syntax.

JSON Arrays
Values in JSON can be arrays.
Example
{
"employees":["John", "Anna", "Peter"]
}

JSON Booleans
Values in JSON can be true/false.
Example
{"sale":true}

JSON null
Values in JSON can be null.
Example
{"middlename":null}

JSON.parse()

A common use of JSON is to exchange data to/from a web server.When receiving data from a web server, the data is always a string.Parse the data with JSON.parse(), and the data becomes a JavaScript object.

Example - Parsing JSON
Imagine we received this text from a web server:
'{"name":"John", "age":30, "city":"New York"}'
Use the JavaScript function JSON.parse() to convert text into a JavaScript object:
const obj = JSON.parse('{"name":"John", "age":30, "city":"New York"}');
Make sure the text is in JSON format, or else you will get a syntax error.
Use the JavaScript object in your page:
Example
* *
<script>
document.getElementById("demo").innerHTML = obj.name;
</script>

Array as JSON
When using the JSON.parse() on a JSON derived from an array, the method will return a JavaScript array, instead of a JavaScript object.
Example
const text = '["Ford", "BMW", "Audi", "Fiat"]';
const myArr = JSON.parse(text);

Exceptions
Parsing Dates
Date objects are not allowed in JSON.
If you need to include a date, write it as a string.
You can convert it back into a date object later:

Example
Convert a string into a date:
const text = '{"name":"John", "birth":"1986-12-14", "city":"New York"}';
const obj = JSON.parse(text);
obj.birth = new Date(obj.birth);
document.getElementById("demo").innerHTML = obj.name + ", " + obj.birth;

Parsing Functions
Functions are not allowed in JSON.
If you need to include a function, write it as a string.
You can convert it back into a function later:
Example
Convert a string into a function:
const text = '{"name":"John", "age":"function () {return 30;}", "city":"New York"}';
const obj = JSON.parse(text);
obj.age = eval("(" + obj.age + ")");
document.getElementById("demo").innerHTML = obj.name + ", " + obj.age();

You should avoid using functions in JSON, the functions will lose their scope, and you would have to use eval() to convert them back into functions.

JSON.stringify()

A common use of JSON is to exchange data to/from a web server.When sending data to a web server, the data has to be a string.Convert a JavaScript object into a string with JSON.stringify().

Stringify a JavaScript Object
Imagine we have this object in JavaScript:

const obj = {name: "John", age: 30, city: "New York"};

Use the JavaScript function JSON.stringify() to convert it into a string.

const myJSON = JSON.stringify(obj);

The result will be a string following the JSON notation.
myJSON is now a string, and ready to be sent to a server:
Example
const obj = {name: "John", age: 30, city: "New York"};
const myJSON = JSON.stringify(obj);

You will learn how to send JSON to a server in the next chapters.

Stringify a JavaScript Array
It is also possible to stringify JavaScript arrays:
Imagine we have this array in JavaScript:

const arr = ["John", "Peter", "Sally", "Jane"];

Use the JavaScript function JSON.stringify() to convert it into a string.

const myJSON = JSON.stringify(arr);

The result will be a string following the JSON notation.
myJSON is now a string, and ready to be sent to a server:
Example
const arr = ["John", "Peter", "Sally", "Jane"];
const myJSON = JSON.stringify(arr);

Storing Data
When storing data, the data has to be a certain format, and regardless of where you choose to store it, *text* is always one of the legal formats.
JSON makes it possible to store JavaScript objects as text.
Example
Storing data in local storage
```
// Storing data:
const myObj = {name: "John", age: 31, city: "New York"};
const myJSON = JSON.stringify(myObj);
localStorage.setItem("testJSON", myJSON);

// Retrieving data:
let text = localStorage.getItem("testJSON");
let obj = JSON.parse(text);
document.getElementById("demo").innerHTML = obj.name;
```

W 4 D 2

JS OBJECTS (JavaScript Objects)

Introduction to the JavaScript objects :
In JavaScript, an object is an unordered collection of key-value pairs. Each key-value pair is called a property.The key of a property can be a string. And the value of a property can be any value, e.g., a string, a number, an array, and even a function.
JavaScript provides you with many ways to create an object. The most commonly used one is to use the object literal notation.
The following example creates an empty object using the object literal notation:
```
let empty = {};
```
To create an object with properties, you use the key:value within the curly braces. For example, the following creates a new person object:
```
let person = {
firstName: 'John',
lastName: 'Doe'
};
```
The person object has two properties firstName and lastName with the corresponding values 'John' and 'Doe'.
When an object has multiple properties, you use a comma (,) to separate them like the above example.

Introduction to the JavaScript object methods

An object is a collection of key/value pairs or properties. When the value is a function, the property becomes a method. Typically, you use methods to describe the object behaviors.

For example, the following adds the greet method to the person object:

```
let person = {
firstName: 'John',
lastName: 'Doe'
```

```
};

person.greet = function () {
console.log('Hello!');
}

person.greet();
```

Output:
Hello!

In this example:

- First, use a function expression to define a function and assign it to the greet property of the person object.
- Then, call the method greet() method.

Besides using a function expression, you can define a function and assign it to an object like this:

```
let person = {
firstName: 'John',
lastName: 'Doe'
};
function greet() {
console.log('Hello, World!');
}
person.greet = greet;
person.greet();
```

Introduction to JavaScript constructor functions

In the JavaScript objects tutorial, you learned how to use the object literal syntax to create a new object.

For example, the following creates a new person object with two properties firstName and lastName:

```
let person = {
firstName: 'John',
lastName: 'Doe'
};
```

In practice, you often need to create many similar objects like the person object.

To do that, you can use a constructor function to define a custom type and the new operator to create multiple objects from this type.

Technically speaking, a constructor function is a regular function with the following convention:

- The name of a constructor function starts with a capital letter like Person, Document, etc.
- A constructor function should be called only with the new operator.

The following example defines a constructor function called Person:

```
function Person(firstName, lastName) {
this.firstName = firstName;
this.lastName = lastName;
}
```

In this example, the Person is the same as a regular function except that its name starts with the capital letter P.

To create a new instance of the Person, you use the new operator:

```
let person = new Person('John','Doe');
```

Basically, the new operator does the following:

- Create a new empty object and assign it to the this variable.
- Assign the arguments 'John' and 'Doe' to the firstName and lastName properties of the object.
- Return the this value.

It's functionally equivalent to the following:

```
function Person(firstName, lastName) {
   // this = {};
   // add properties to this
this.firstName = firstName;
this.lastName = lastName;
   // return this;
}
```

Therefore, the following statement:

```
let person = new Person('John','Doe');
```

… returns the same result as the following statement:

```
let person = {
firstName: 'John',
lastName: 'Doe'
};
```

However, the constructor function Person allows you to create multiple similar objects. For example:

```
let person1 = new Person('Jane','Doe')
let person2 = new Person('James','Smith')
```

Adding methods to JavaScript constructor functions

An object may have methods that manipulate its data. To add a method to an object created via the constructor function, you can use the this keyword. For example:

```
function Person(firstName, lastName) {
this.firstName = firstName;
this.lastName = lastName;
```

```
this.getFullName = function () {
returnthis.firstName + " " + this.lastName;
   };
}
```

Now, you can create a new Person object and invoke the getFullName() method:

```
let person = new Person("John", "Doe");
console.log(person.getFullName());
```

Output:
John Doe

The problem with the constructor function is that when you create multiple instances of the Person, the this.getFullName() is duplicated in every instance, which is not memory efficient.

To resolve this, you can use the prototype so that all instances of a custom type can share the same methods.

Returning from constructor functions

Typically, a constructor function implicitly returns this that set to the newly created object. But if it has a return statement, then here's are the rules:

- If return is called with an object, the constructor function returns that object instead of this.
- If return is called with a value other than an object, it is ignored.

Calling a constructor function without the new keyword

Technically, you can call a constructor function like a regular function without using the new keyword like this:

```
let person = Person('John','Doe');
```

In this case, the Person just executes like a regular function. Therefore, the this inside the Person function doesn't bind to the person variable but the global object.

If you attempt to access the firstName or lastName property, you'll get an error:

```
console.log(person.firstName);
```

Error:
TypeError: Cannot read property 'firstName' of undefined

Similarly, you cannot access the getFullName() method since it's bound to the global object.

```
person.getFullName();
```

Error:
TypeError: Cannot read property 'getFullName' of undefined

To prevent a constructor function to be invoked without the new keyword, ES6 introduced the new.target property.

If a constructor function is called with the new keyword, the new.target returns a reference of the function. Otherwise, it returns undefined.

The following adds a statement inside the Person function to show the new.target to the console:

```
function Person(firstName, lastName)
{
console.log(new.target);

this.firstName = firstName;
this.lastName  =lastName;

this.getFullName = function ()
{
returnthis.firstName + " " + this.lastName;
   };
}
```

The following returns undefined because the Person constructor function is called like a regular function:

```
let person = Person("John", "Doe");
```

Output:
undefined

However, the following returns a reference to the Person function because it's called with the new keyword:

```
let person = new Person("John", "Doe");
```

Output:
[Function: Person]

By using the new.target, you can force the callers of the constructor function to use the new keyword. Otherwise, you can throw an error like this:

```
function Person(firstName, lastName) {
if (!new.target) {
throw Error("Cannot be called without the new keyword");
   }
this.firstName = firstName;
this.lastName = lastName;
}
```

Alternatively, you can make the syntax more flexible by creating a new Person object if the users of the constructor function don't use the new keyword:

```
function Person(firstName, lastName)
{
if (!new.target)
 {
return new Person(firstName, lastName);
```

```
    }
this.firstName = firstName;
this.lastName = lastName;
}
let person = Person("John", "Doe");
console.log(person.firstName);
```

Accessing properties

To access a property of an object, you use one of two notations: the dot notation and array-like notation.

1) The dot notation (.)

The following illustrates how to use the dot notation to access a property of an object:

objectName.propertyName

For example, to access the firstName property of the person object, you use the following expression:

person.firstName

This example creates a person object and shows the first name and last name to the console:

```
let person = {
firstName: 'John',
lastName: 'Doe'
};
console.log(person.firstName);
console.log(person.lastName);
```

2) Array-like notation ([])

The following illustrates how to access the value of an object's property via the array-like notation:

objectName['propertyName']

For example:

```
let person = {
firstName: 'John',
lastName: 'Doe'
};
console.log(person['firstName']);
console.log(person['lastName']);
```

When a property name contains spaces, you need to place it inside quotes. For example, the following address object has the 'building no' as a property:

```
let address = {
    'building no': 3960,
street: 'North 1st street',
state: 'CA',
country: 'USA'
};
```

To access the 'building no' property, you need to use the array-like notation:

address['building no'];

If you use the dot notation, you'll get an error:

address.'building no';

Error:

SyntaxError: Unexpected string

Note that it is not a good practice to use spaces in the property names of an object.

Reading from a property that does not exist will result in an undefined. For example:

console.log(address.district);

Output:

undefined

Modifying the value of a property

To change the value of a property, you use the assignment operator (=). For example:

```
let person = {
firstName: 'John',
lastName: 'Doe'
};
person.firstName = 'Jane';
console.log(person);
```

Output:

{ firstName: 'Jane', lastName: 'Doe' }

In this example, we changed the value of the firstName property of the person object from 'John' to 'Jane'.

Adding a new property to an object

Unlike objects in other programming languages such as Java and C#, you can add a property to an object after object creation.

The following statement adds the age property to the person object and assigns 25 to it:

person.age = 25;

Deleting a property of an object

To delete a property of an object, you use the delete operator:

deleteobjectName.propertyName;

The following example removes the age property from the person object:

deleteperson.age;

If you attempt to reaccess the age property, you'll get an undefined value.

Objects have two types of properties: data and accessor properties.

1) Data properties

A data property contains a single location for a data value. A data property has four attributes:

- [[Configurarable]] – determines whether a property can be redefined or removed via delete operator.
- [[Enumerable]] – indicates if a property can be returned in the for...in loop.
- [[Writable]] – specifies that the value of a property can be changed.
- [[Value]] – contains the actual value of a property.

By default, the [[Configurable]] , [[Enumerable]]And [[Writable]] attributes set to true for all properties defined directly on an object. The default value of the[[Value]] attribute is undefined.

The following example creates a person object with two properties firstName and lastName with the configurable, enumerable, and writable attributes set to true. And their values are set to 'John' and 'Doe' respectively:

```
let person = {
firstName: 'John',
lastName: 'Doe'
};
```

To change any attribute of a property, you use the Object.defineProperty() method.
The Object.defineProperty() method accepts three arguments:

- An object.
- A property name of the object.
- A property descriptor object that has four properties: configurable, enumerable, writable, and value.

If you use the Object.defineProperty() method to define a property of the object, the default values of [[Configurable]], [[Enumerable]], and [[Writable]] are set to false unless otherwise specified.
The following example creates a person object with the age property:

```
let person = {};
person.age = 25;
```

Since the default value of the [[Configurable]] attribute is set to true, you can remove it via the delete operator:

```
deleteperson.age;
console.log(person.age);
```

Output:
undefined

The following example creates a person object and adds the ssn property to it using the Object.defineProperty() method:

```
'use strict';

let person = {};

Object.defineProperty(person, 'ssn', {
configurable: false,
value: '012-38-9119'
});

deleteperson.ssn;
```

Output:
TypeError: Cannot delete property 'ssn' of #<Object>

In this example, the configurable attribute is set to false. herefore, deleting the ssn property causes an error.
Also, once you define a property as non-configurable, you cannot change it to configurable.
If you use the Object.defineProperty() method to change any attribute other than the writable, you'll get an error. or example:

```
'use strict';

let person = {};

Object.defineProperty(person, 'ssn', {
configurable: false,
value: '012-38-9119'
});


Object.defineProperty(person, 'ssn', {
configurable: true
});
```

Code language: JavaScript (javascript)
Output:
TypeError: Cannot redefine property: ssn

By default, the enumerable attribute of all the properties defined on an object is true. t means that you can iterate over all object properties using the for...in loop like this:

```
let person = {};
person.age = 25;
person.ssn = '012-38-9119';

for (let property in person) {
console.log(property);
}
```

Output:
age
ssn

The following makes the ssn property non-enumerable by setting the enumerable attribute to false.

```
let person = {};
person.age = 25;
person.ssn = '012-38-9119';

Object.defineProperty(person, 'ssn', {
enumerable: false
});

for (let prop in person) {
console.log(prop);
}
```

Output
Age

2) Accessor properties

Similar to data properties, accessor properties also have [[Configurable]] and [[Enumerable]] attributes. But the accessor properties have the [[Get]] and [[Set]] attributes instead of [[Value]] and [[Writable]].
When you read data from an accessor property, the [[Get]] function is called automatically to return a value. The default return value of the [[Get]] function is undefined.
If you assign a value to an accessor property, the [[Set]] function is called automatically.
To define an accessor property, you must use the Object.defineProperty() method. or example:

```
let person = {
firstName: 'John',
lastName: 'Doe'
}

Object.defineProperty(person, 'fullName', {
get: function () {
returnthis.firstName + ' ' + this.lastName;
   },
set: function (value) {
let parts = value.split(' ');
if (parts.length == 2) {
this.firstName = parts[0];
this.lastName = parts[1];
      } else {
throw 'Invalid name format';
      }
   }
});
console.log(person.fullName);
```

Output:
'John Doe'

In this example:

- First, define the person object that contains two properties: firstName and lastName.
- Then, add the fullName property to the person object as an accessor property.

In the fullname accessor property:

- The [[Get]] returns the full name that is the result of concatenating of firstName, space, and lastName.
- The [[Set]] method splits the argument by the space and assigns the firstName and lastName properties the corresponding parts of the name.
- If the full name is not in the correct format i.e., first name, space, and last name, it will throw an error.

Define multiple properties: Object.defineProperties()
In ES5, you can define multiple properties in a single statement using the Object.defineProperties() method. or example:

```
var product = {};

Object.defineProperties(product, {
name: {
```

```
value: 'Smartphone'
   },
price: {
value: 799
   },
tax: {
value: 0.1
   },
netPrice: {
get: function () {
returnthis.price * (1 + this.tax);
      }
   }
});

console.log('The net price of a ' + product.name + ' is ' + product.netPrice.toFixed(2) + ' USD');
```

Output:
The net price of a Smartphone is 878.90 USD

In this example, we defined three data properties: name, price, and tax, and one accessor property netPrice for the product object.

JavaScript object property descriptor

The Object.getOwnPropertyDescriptor() method allows you to get the descriptor object of a property. The Object.getOwnPropertyDescriptor() method takes two arguments:

1. An object
2. A property of the object

It returns a descriptor object that describes a property. The descriptor object has four properties: configurable, enumerable, writable, and value.

The following example gets the descriptor object of the name property of the product object in the prior example.

```
let person = {
firstName: 'John',
lastName: 'Doe'
};

let descriptor = Object.getOwnPropertyDescriptor(person, 'firstName');

console.log(descriptor);
```

Output:

```
{ value: 'John',
writable: true,
enumerable: true,
configurable: true }
```

Introduction to JavaScript prototype

In JavaScript, every function and object has a property named prototype by default. For example,

```
function Person () {
    this.name = 'John',
this.age = 23
}
constman = new Person();

// checking the prototype value
console.log(Person.prototype); // { ... }
```

Output :
Person {}

In the above example, we are trying to access the prototype property of a Person constructor function. Since the prototype property has no value at the moment, it shows an empty object { ... }.

Prototype Inheritance

In JavaScript, a prototype can be used to add properties and methods to a constructor function. And objects inherit properties and methods from a prototype. For example,

```
// constructor function
function Person () {
    this.name = 'John',
this.age = 23
}

// creating objects
const person1 = new Person();
const person2 = new Person();

// adding property to constructor function
Person.prototype.gender = 'male';

// prototype value of Person
console.log(Person.prototype);

// inheriting the property from prototype
console.log(person1.gender);
console.log(person2.gender);
```

Output
{ gender: "male" }
male
male

In the above program, we have added a new property gender to the Person constructor function using:

Person.prototype.gender = 'male';
Then object person1 and person2 inherits the property gender from the prototype property of Person constructor function.
Hence, both objects person1 and person2 can access the gender property.
Note: The syntax to add the property to an object constructor function is:
objectConstructorName.prototype.key = 'value';

Prototype is used to provide additional property to all the objects created from a constructor function.

Add Methods to a Constructor Function Using Prototype

You can also add new methods to a constructor function using prototype. For example,

```
// constructor function
function Person () {
    this.name = 'John',
this.age = 23
}
// creating objects
const person1 = new Person();
const person2 = new Person();
// adding a method to the constructor function
Person.prototype.greet = function() {
console.log('hello' + ' ' +  this.name);
}
person1.greet(); // hello John
person2.greet(); // hello John
```

Output
hello John
hello John
In the above program, a new method greet is added to the Person constructor function using a prototype.

In JavaScript, objects can inherit features from one another via prototypes. Every object has its own property called prototype.Because a prototype itself is also another object, the prototype has its own prototype. This creates a something called prototype chain. The prototype chain ends when a prototype has null for its own prototype.

W4 D3

ES6

What is ES6?
ES6 stands for ECMAScript 6.(ECMA-**European Computer Manufacturers Association)**
ECMAScript was created to standardize JavaScript, and ES6 is the 6th version of ECMAScript, it was published in 2015, and is also known as ECMAScript 2015.

Browser Support for ES6 (2015)

Safari 10 and Edge 14 were the first browsers to fully support ES6:

Chrome 58	Edge 14	Firefox 54	Safari 10	Opera 55
Jan 2017	Aug 2016	Mar 2017	Jul 2016	Aug 2018

1.Arrow Functions

Arrow functions are introduced in ES6, which provides you a more accurate way to write the functions in JavaScript. They allow us to write smaller function syntax. Arrow functions make your code more readable and structured.

Arrow functions are **anonymous functions** (the functions without a name and not bound with an identifier). They don't return any value and can declare without the function keyword. Arrow functions cannot be used as the constructors. The context within the arrow functions is lexically or statically defined. They are also called as **Lambda Functions** in different languages.

Arrow functions do not include any prototype property, and they cannot be used with the new keyword.

Syntax for defining the arrow function

```
const functionName = (arg1, arg2, ?..) => {
   //body of the function
}
```

There are three parts to an Arrow Function or Lambda Function:

Parameters: Any function may optionally have the parameters.

Fat arrow notation/lambda notation: It is the notation for the **arrow (=>).**

Statements: It represents the instruction set of the function.

Let us try to understand it with an example.

In the following example, we are defining three functions that show **Function Expression, Anonymous Function, and Arrow Function.**

```
// function expression

var myfun1 = function show() {
 console.log("It is a Function Expression");
}

// Anonymous function

var myfun2 = function () {
   console.log("It is an Anonymous Function");
  }
```

```
//Arrow function

var myfun3 = () => {
   console.log("It is an Arrow Function");
  };

myfun1();
myfun2();
myfun3();
```

Output

```
It is a Function Expression
It is an Anonymous Function
It is an Arrow Function
```

2. Template Strings / Literals

Template literals are a new feature introduced in ECMAScript 2015/ ES6. It provides an easy way to create multiline strings and perform string interpolation. Template literals are the string literals and allow embedded expressions.

Before ES6, template literals were called as **template strings**. Unlike quotes in strings, template literals are enclosed by the **backtick (` `)** character (key below the **ESC** key in QWERTY keyboard). Template literals can contain placeholders, which are indicated by the dollar sign and curly braces **($(expression})**. Inside the backticks, if we want to use an expression, then we can place that expression in the **($(expression})**.

Syntax

```
var str = `string value`;
```

Multiline strings

In normal strings, we have to use an escape sequence **\n** to give a new line for creating a multiline string. However, in template literals, there is no need to use **\n** because string ends only when it gets **backtick(`)** character.

Let us try to understand it with the following example.

Example

```
// Without template literal
console.log('Without template literal \n multiline string');

// With template literal
console.log(`Using template literal
multiline string`);
```

Output

```
Without template literal
multiline string
Using template literal
multiline string
```

String Interpolation

ES6 template literals support string interpolation. Template literals can use the placeholders for string substitution. To embed expressions with normal strings, we have to use the **${}** syntax.

Example -1

```
var name = 'World';
var cname = 'javaTpoint';
```

```
console.log(`Hello, ${name}!
Welcome to ${cname}`);
```

Output

```
Hello, World!
Welcome to javaTpoint
```

3.Spread Operator

The JavaScript spread operator (...) allows us to quickly copy all or part of an existing array or object into another array or object.

Example

```
constnumbersOne = [1, 2, 3];
constnumbersTwo = [4, 5, 6];
constnumbersCombined = [...numbersOne, ...numbersTwo];
```

Html Code

```
<!DOCTYPE html>
<html>
<body>

<script>
constnumbersOne = [1, 2, 3];
constnumbersTwo = [4, 5, 6];
constnumbersCombined = [...numbersOne, ...numbersTwo];
document.write(numbersCombined);
</script>
</body>
</html>
```

Output :

1,2,3,4,5,6

We can use the spread operator with objects too:

Example

Combine these two objects:

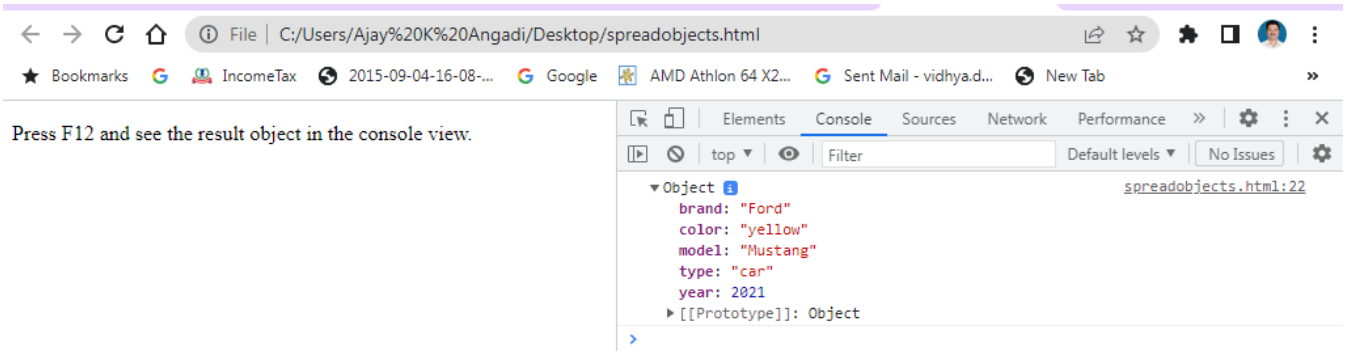
```
constmyVehicle = {
brand: 'Ford',
model: 'Mustang',
color: 'red'
}
constupdateMyVehicle = {
type: 'car',
year: 2021,
color: 'yellow'
}
constmyUpdatedVehicle = {...myVehicle, ...updateMyVehicle}
```

Html Code :

```
<!DOCTYPE html>
<html>
```

```
<body>
<script>
constmyVehicle = {
brand: 'Ford',
model: 'Mustang',
color: 'red'
}

constupdateMyVehicle = {
type: 'car',
year: 2021,
color: 'yellow'
}
constmyUpdatedVehicle = {...myVehicle, ...updateMyVehicle}
//Check the result object in the console:
console.log(myUpdatedVehicle);
</script>
<p>Press F12 and see the result object in the console view.</p>
</body>
</html>
```

Output :



4.ES6 Map

ES6 is a series of new features that are added to the JavaScript. Prior to ES6, when we require the mapping of keys and values, we often use an object. It is because the object allows us to map a key to the value of any type.

ES6 provides us a new collection type called **Map**, which holds the key-value pairs in which values of any type can be used as either keys or values. A Map object always remembers the actual insertion order of the keys. Keys and values in a Map object may be primitive or objects. It returns the new or empty Map.

Maps are ordered, so they traverse the elements in their insertion order.

Syntax

For creating a new Map, we can use the following syntax:

*var map = **new** Map([iterable]);*

The **Map ()** accepts an optional iterable object, whose elements are in the key-value pairs.

Map Properties

S.no.	Properties	Description
1.	**Map.prototype.size**	This property returns the number of key-value pairs in the Map object.

Let us understand the above property of Map object in brief.

Map.prototype.size

It returns the number of elements in the Map object.

Syntax

Map.size

Example

```
var map = new Map();
   map.set('John', 'author');
   map.set('arry', 'publisher');
   map.set('Mary', 'subscriber');
   map.set('James', 'Distributor');
console.log(map.size);
```

Output

```
4
```

5.S6 Set

A set is a data structure that allows you to create a collection of unique values.Set is the collection of values similar to arrays, but it does not contain any duplicates. It allows us to store unique values. It supports both primitive values and object references.

As similar to maps, sets are also ordered, i.e., the elements in sets are iterated in their insertion order. It returns the set object.

Syntax

```
var s = new Set("val1","val2","val3");
```

Let us understand the concept of **set** by using the following example:

Example

```
let colors = new Set(['Green', 'Red', 'Orange', 'Yellow', 'Red']);
console.log(colors);
```

All the elements of a Set must be unique. Therefore the set **colors** in the above example only contain four distinct elements. We will get the following output after the successful execution of the above code.

Output

```
Set { 'Green', 'Red', 'Orange', 'Yellow' }
```

Let us see the properties and methods of the Set.

Set Properties

S.no.	Properties	Description
1.	**Set.size**	This property returns the number of values in the set object.

Set.size

This property of the Set object returns the value that represents the number of elements in the Set object.

Example

```
let colors = new Set(['Green', 'Red', 'Orange', 'Yellow', 'Red']);
console.log(colors.size);
console.log(colors);
```

Output

```
4
Set { 'Green', 'Red', 'Orange', 'Yellow' }
```

W6 D4

Servlets – Overview

What are Servlets?

Java Servlets are programs that run on a Web or Application server and act as a middle layer between a requests coming from a Web browser or other HTTP client and databases or applications on the HTTP server.

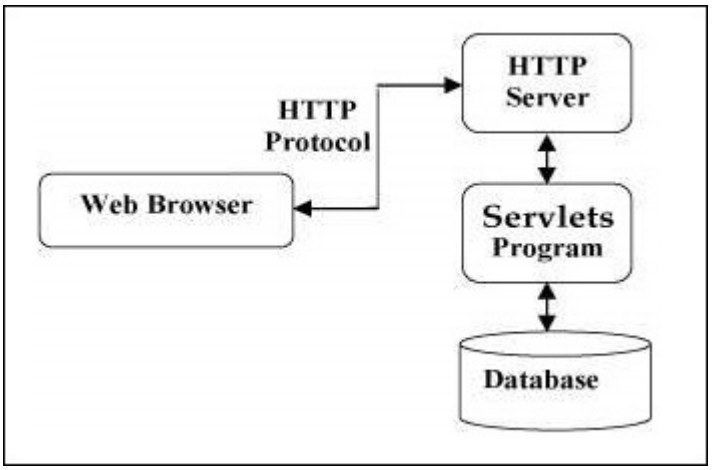
Using Servlets, you can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically.

Java Servlets often serve the same purpose as programs implemented using the Common Gateway Interface (CGI). But Servlets offer several advantages in comparison with the CGI.

- Performance is significantly better.
- Servlets execute within the address space of a Web server. It is not necessary to create a separate process to handle each client request.
- Servlets are platform-independent because they are written in Java.
- Java security manager on the server enforces a set of restrictions to protect the resources on a server machine. So servlets are trusted.
- The full functionality of the Java class libraries is available to a servlet. It can communicate with applets, databases, or other software via the sockets and RMI mechanisms that you have seen already.

Servlets Architecture

The following diagram shows the position of Servlets in a Web Application.



Servlets Tasks

Servlets perform the following major tasks −

- Read the explicit data sent by the clients (browsers). This includes an HTML form on a Web page or it could also come from an applet or a custom HTTP client program.
- Read the implicit HTTP request data sent by the clients (browsers). This includes cookies, media types and compression schemes the browser understands, and so forth.
- Process the data and generate the results. This process may require talking to a database, executing an RMI or CORBA call, invoking a Web service, or computing the response directly.
- Send the explicit data (i.e., the document) to the clients (browsers). This document can be sent in a variety of formats, including text (HTML or XML), binary (GIF images), Excel, etc.
- Send the implicit HTTP response to the clients (browsers). This includes telling the browsers or other clients what type of document is being returned (e.g., HTML), setting cookies and caching parameters, and other such tasks.

Servlets - Life Cycle

A servlet life cycle can be defined as the entire process from its creation till the destruction. The following are the paths followed by a servlet.

- The servlet is initialized by calling the init() method.
- The servlet calls service() method to process a client's request.
- The servlet is terminated by calling the destroy() method.
- Finally, servlet is garbage collected by the garbage collector of the JVM.

Now let us discuss the life cycle methods in detail.

The init() Method

The init method is called only once. It is called only when the servlet is created, and not called for any user requests afterwards. So, it is used for one-time initializations, just as with the init method of applets.

The servlet is normally created when a user first invokes a URL corresponding to the servlet, but you can also specify that the servlet be loaded when the server is first started.

When a user invokes a servlet, a single instance of each servlet gets created, with each user request resulting in a new thread that is handed off to doGet or doPost as appropriate. The init() method simply creates or loads some data that will be used throughout the life of the servlet.

The init method definition looks like this −

```
public void init() throws ServletException {
  // Initialization code...
}
```

The service() Method

The service() method is the main method to perform the actual task. The servlet container (i.e. web server) calls the service() method to handle requests coming from the client(browsers) and to write the formatted response back to the client.

Each time the server receives a request for a servlet, the server spawns a new thread and calls service. The service() method checks the HTTP request type (GET, POST, PUT, DELETE, etc.) and calls doGet, doPost, doPut, doDelete, etc. methods as appropriate.

Here is the signature of this method −

```
public void service(ServletRequest request, ServletResponse response)
throwsServletException, IOException {
}
```

The service () method is called by the container and service method invokes doGet, doPost, doPut, doDelete, etc. methods as appropriate. So you have nothing to do with service() method but you override either doGet() or doPost() depending on what type of request you receive from the client.

The doGet() and doPost() are most frequently used methods with in each service request. Here is the signature of these two methods.

The doGet() Method

A GET request results from a normal request for a URL or from an HTML form that has no METHOD specified and it should be handled by doGet() method.

```
public void doGet(HttpServletRequest request, HttpServletResponse response)
throwsServletException, IOException {
  // Servlet code
}
```

The doPost() Method

A POST request results from an HTML form that specifically lists POST as the METHOD and it should be handled by doPost() method.

```
public void doPost(HttpServletRequest request, HttpServletResponse response)
throwsServletException, IOException {
  // Servlet code
}
```

The destroy() Method

The destroy() method is called only once at the end of the life cycle of a servlet. This method gives your servlet a chance to close database connections, halt background threads, write cookie lists or hit counts to disk, and perform other such cleanup activities.

After the destroy() method is called, the servlet object is marked for garbage collection. The destroy method definition looks like this −

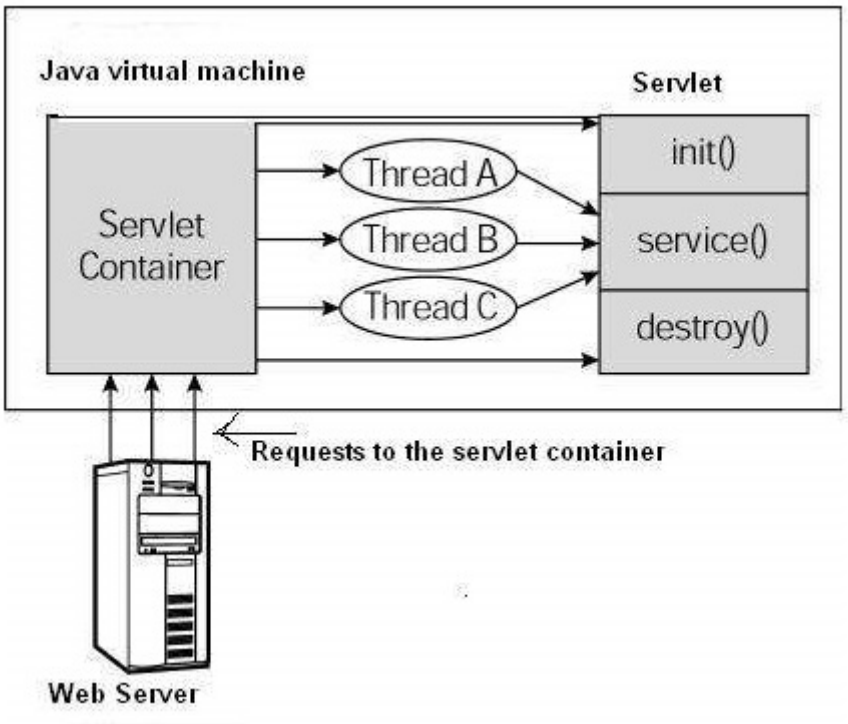
```
public void destroy() {
  // Finalization code...
}
```

Architecture Diagram

The following figure depicts a typical servlet life-cycle scenario.

- First the HTTP requests coming to the server are delegated to the servlet container.
- The servlet container loads the servlet before invoking the service() method.

- Then the servlet container handles multiple requests by spawning multiple threads, each thread executing the service() method of a single instance of the servlet.



Servlets Packages

Java Servlets are Java classes run by a web server that has an interpreter that supports the Java Servlet specification.

Servlets can be created using the javax.servlet and javax.servlet.http packages(jakarta.servlet and jakarta.servlet.http packages in latest JDK versions) which are a standard part of the Java's enterprise edition, an expanded version of the Java class library that supports large-scale development projects.

These classes implement the Java Servlet and JSP specifications. At the time of writing this tutorial, the versions are Java Servlet 2.5 and JSP 2.1.

Java servlets have been created and compiled just like any other Java class. After you install the servlet packages and add them to your computer's Classpath, you can compile servlets with the JDK's Java compiler or any other current compiler.

Servlets - Environment Setup

A development environment is where you would develop your Servlet, test them and finally run them.

Like any other Java program, you need to compile a servlet by using the Java compiler javac and after compilation the servlet application, it would be deployed in a configured environment to test and run..

This development environment setup involves the following steps −

A.Setting up Java Development Kit

Downloading JDK 19 for windows os:

1. Go to any browser any load google search engine
2. Type “oracle java” as search element
3. Click on first link i.e https://www.oracle.com/in/java/
4. Click on “Download java” button at upper right corner.
5. Click on Java 19 tab and select windows operating system.

6. Then select second link “x64 installer”.
7. Open downloaded jdk19 installation file to install java software and change destination folder in C drive as c:\java where jdk 19 will be installed.

B.Setting up Web Server − Tomcat

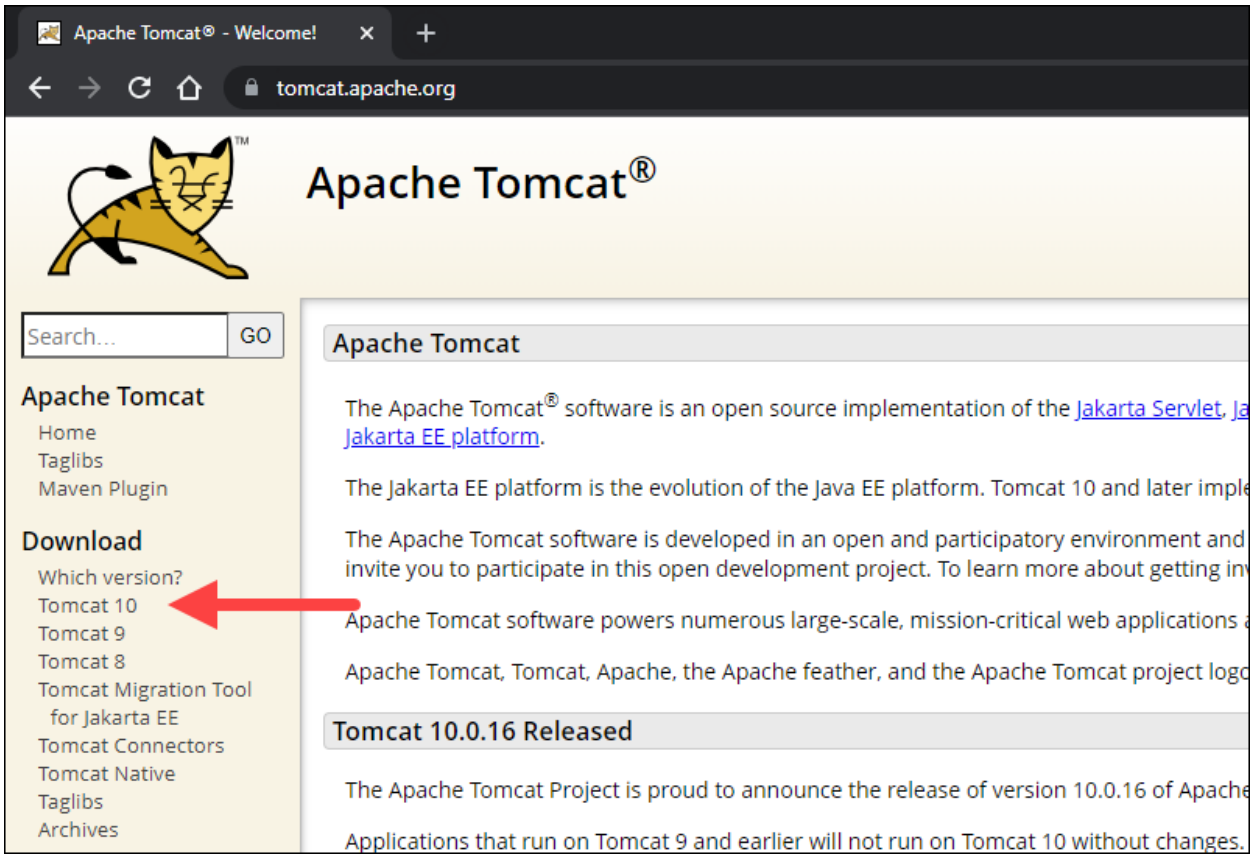
A number of Web Servers that support servlets are available in the market. Some web servers are freely downloadable and Tomcat is one of them.

Apache Tomcat is an open source software implementation of the Java Servlet and Java Server Pages technologies and can act as a standalone server for testing servlets and can be integrated with the Apache Web Server. Here are the steps to setup Tomcat on your machine −

After startup, the default web applications included with Tomcat will be available by visiting http://localhost:8080/. If everything is fine then it should display following result –

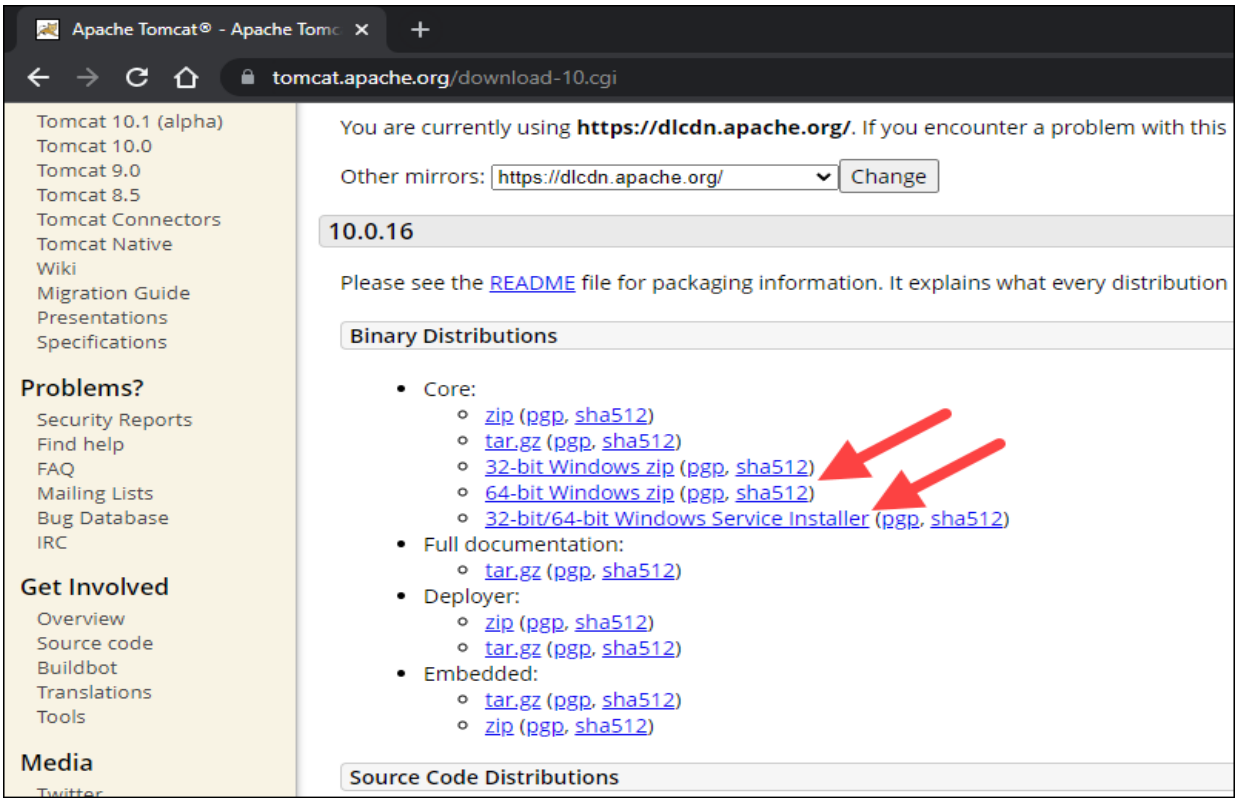
To download the Tomcat installation file, follow the steps below:

1. Browse to the official Apache Tomcat website. Locate the Download section and click the latest Tomcat version available. At the time of writing this article, the latest Tomcat version was version 10.



2. On the Download page, scroll down and locate the Binary Distributions area.

In the Core list, depending on the installation type you prefer, click the download link for the Windows Service Installer or the 32bit/64bit Windows zip file.



Step 2: Install Tomcat

Install Tomcat via the Windows Service Installer for an automated and wizard-guided experience. The service installer installs the Tomcat service and runs it automatically when the system boots.

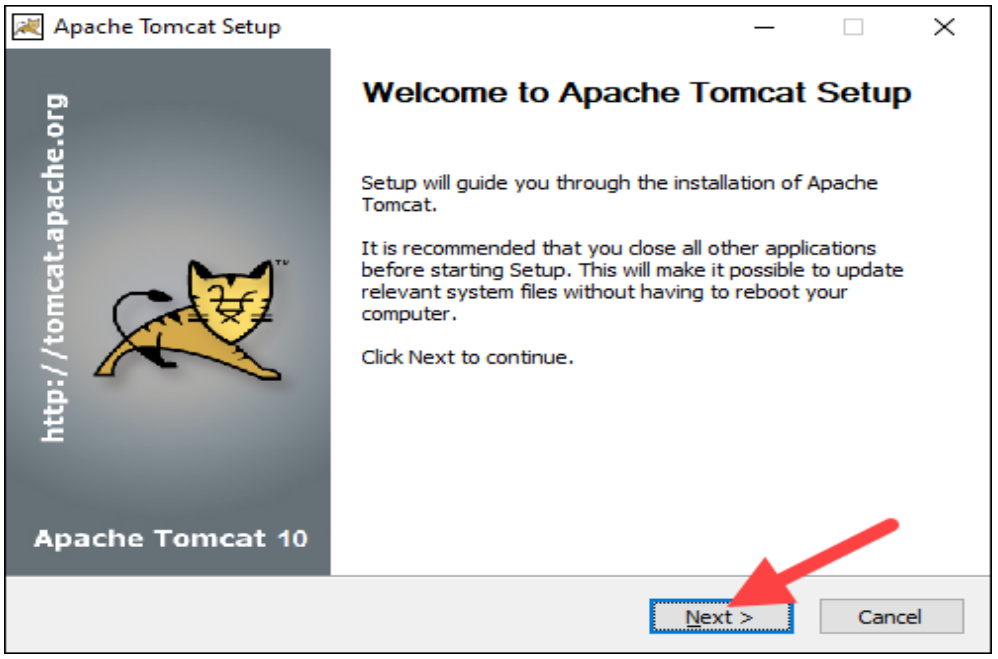
For a portable experience, install Tomcat using the zip file and avoid installing the service. Easily uninstall Tomcat when it is no longer needed by deleting the Tomcat directory, or move it around when necessary.

Install Tomcat Using the Windows Service Installer

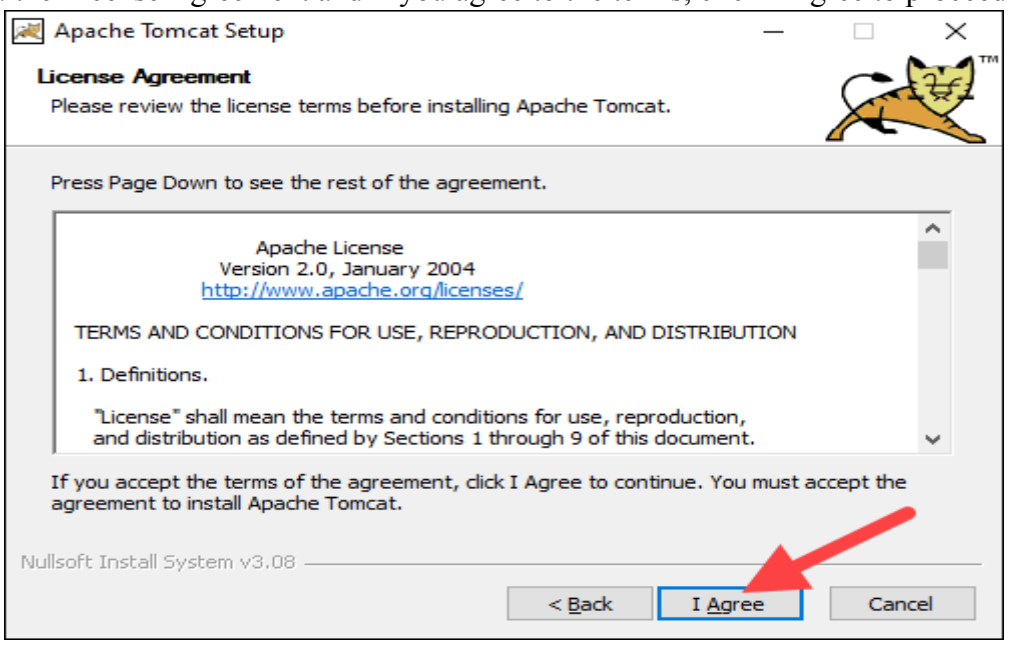
Follow the steps below to install Tomcat using the Windows Service Installer.

1. Open the downloaded Windows Service Installer file to start the installation process.

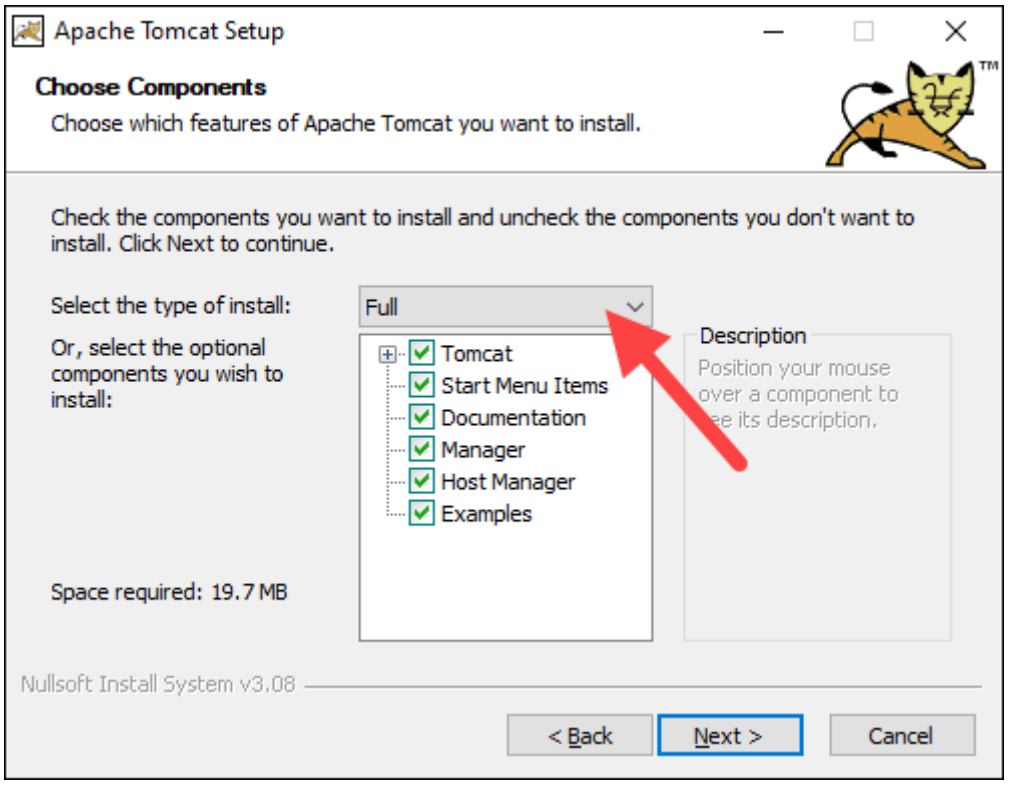
2. In the Tomcat Setup welcome screen, click Next to proceed.



3.Read the License Agreement and if you agree to the terms, click I Agree to proceed to the next step.



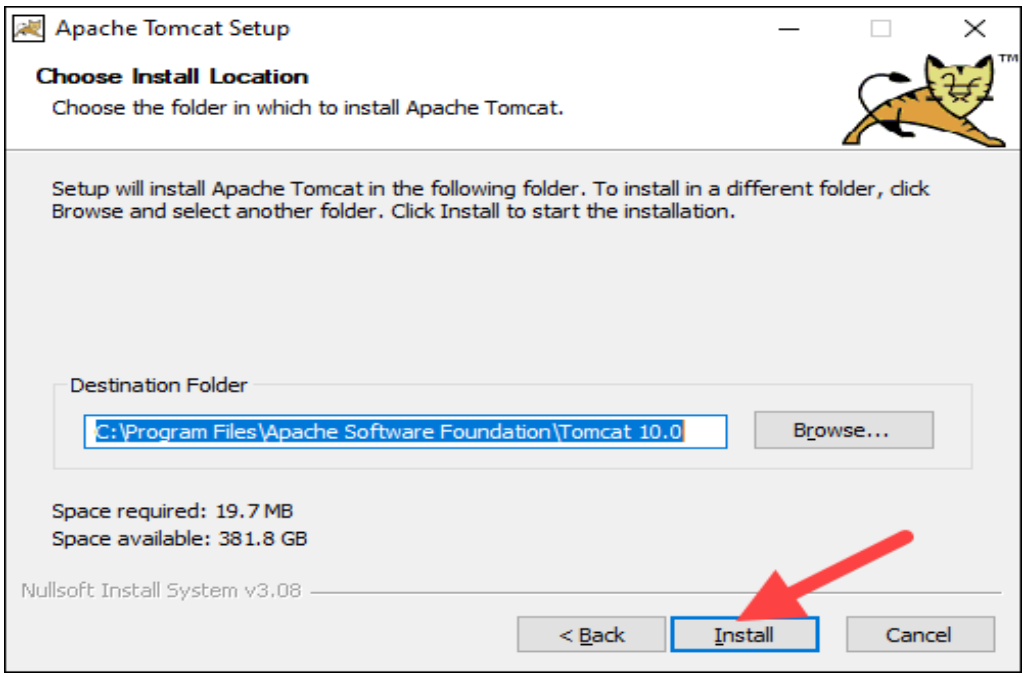
4. In the Tomcat component selection screen, choose Full in the dropdown menu to ensure the wizard installs the Tomcat Host Manager and Servlet and JSP examples web applications. Alternatively, keep the default Normal installation type and click Next.



5. The next step configures the Tomcat server. For instance, enter the Administrator login credentials or choose a different connection port. When finished, click Next to proceed to the next step.

6. The next step requires you to enter the full path to the JRE directory on your system. The wizard auto-completes this if you have previously set up the Java environment variables. Click Next to proceed to the next step.

7. Choose the Tomcat server install location or keep the default one and click Install.



8. Check the Run Apache Tomcat box to start the service after the installation finishes. Optionally, check the Show Readme box to see the Readme file. To complete the installation, click Finish.

9. A popup window appears that starts the Tomcat service. After the process completes, the window closes automatically. The Apache Tomcat web server is now successfully installed .

C. Install and Configure Eclipse IDE to run servlet

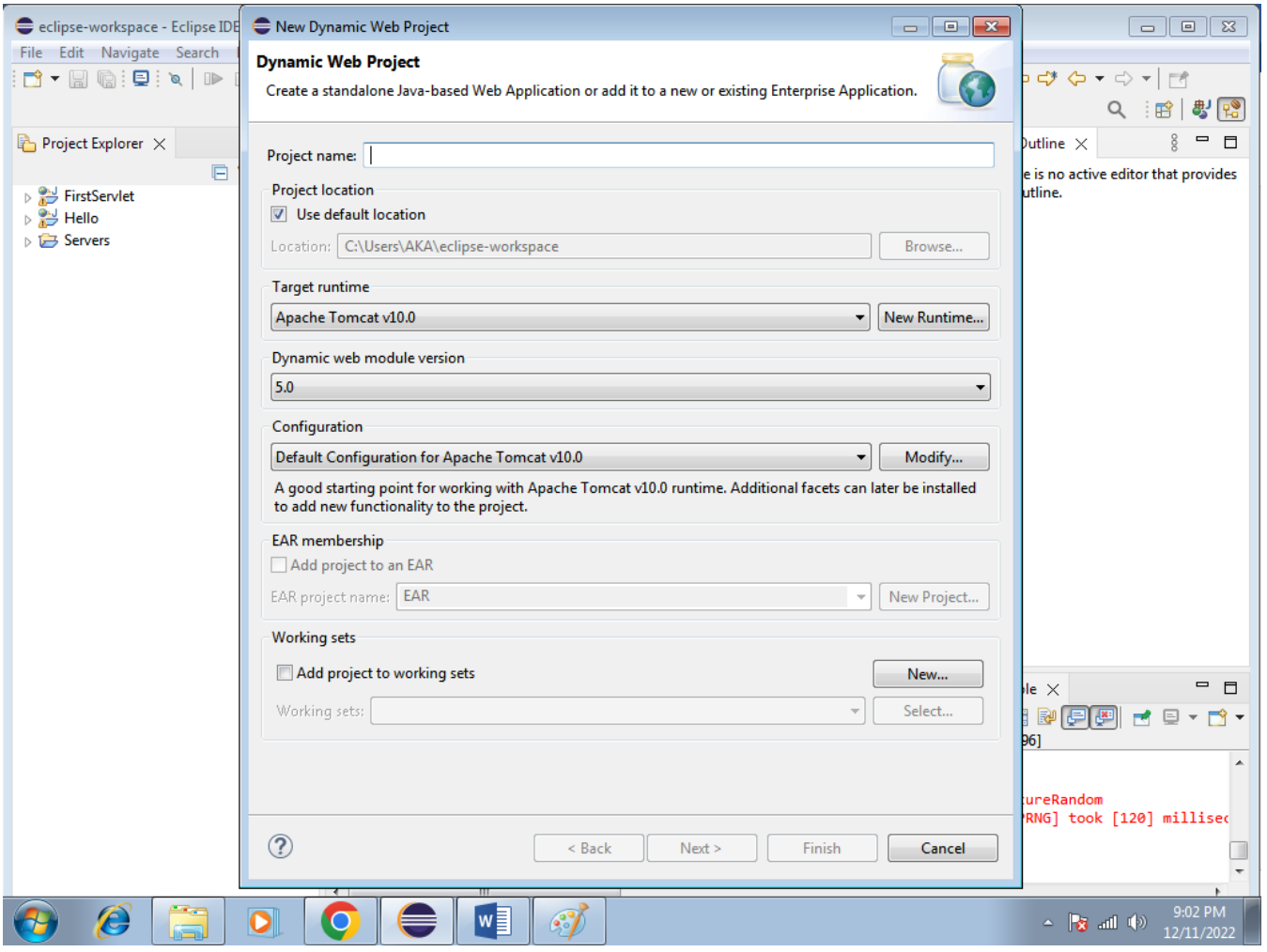
The Eclipse is defined as platform for developing the computer-based applications using various programming language like JAVA, Python, C/C++, Ruby and many more. The Eclipse is IDE (Integrated development kit) and mainly JAVA based programming is done in this platform. There are several plug-ins and other additional plug-ins can be installed in the platform. The advanced client applications can be developed. The JDT is used for doing the programming in Eclipse IDE.

Install Eclipse on Windows

Go to this link https://www.eclipse.org/downloads. Download “Eclipse IDE for Java Developers”. You would see two options on the right side (32 bit and 64 bit), click on 32 bit if you system is 32 bit else click on 64 bit. This will download a zipped file on your system.
To install Eclipse, unzip the downloaded file and copy the unzipped folder to the desired location.

Creating Servlet in Eclipse IDE

Step 1: Create a Project:
Lets create a Servlet application in Eclipse. Open Eclipse and then click File ❯ New ❯ Click Dynamic Web Project.

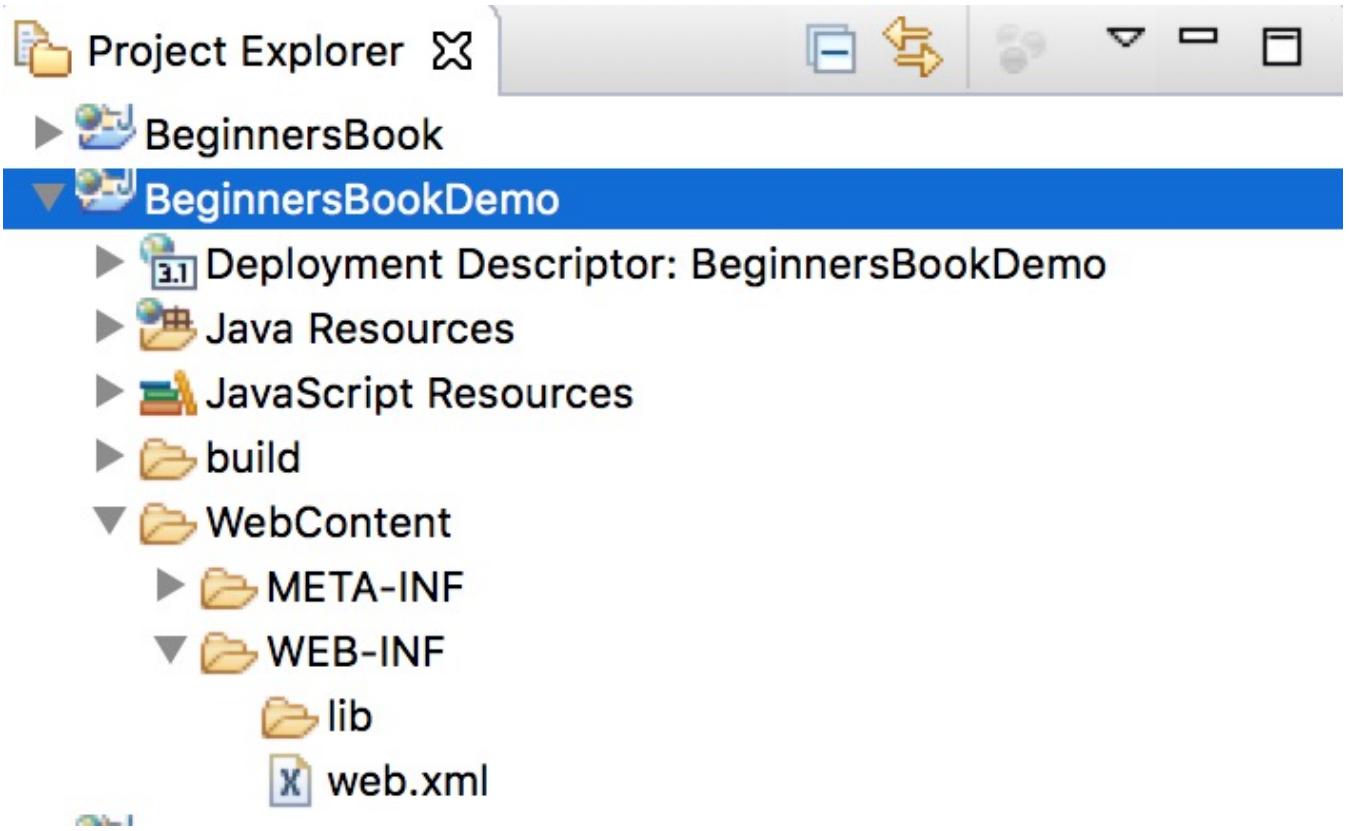


If you do not see dynamic web project option and Servers options in Eclipse then refer this tutorial:

1. Click on Help and then click on “Install New Software”.
2. **In type or select a site option select** https://download/eclipse.org/releases/2022-09\
3. Scroll down to find “**Web, XML, Java EE and OSGI Enterprise Development**” option and expand it.
4. Click next and you would see that the software are installing. Wait for some time and then a popup would ask your permission to restart the Eclipse. Restart it and you would find the dynamic web project option under project list.
5. Goto window option in main eclipse ide =>Perspective=>Open Perspective=>Other and select Java EE option.

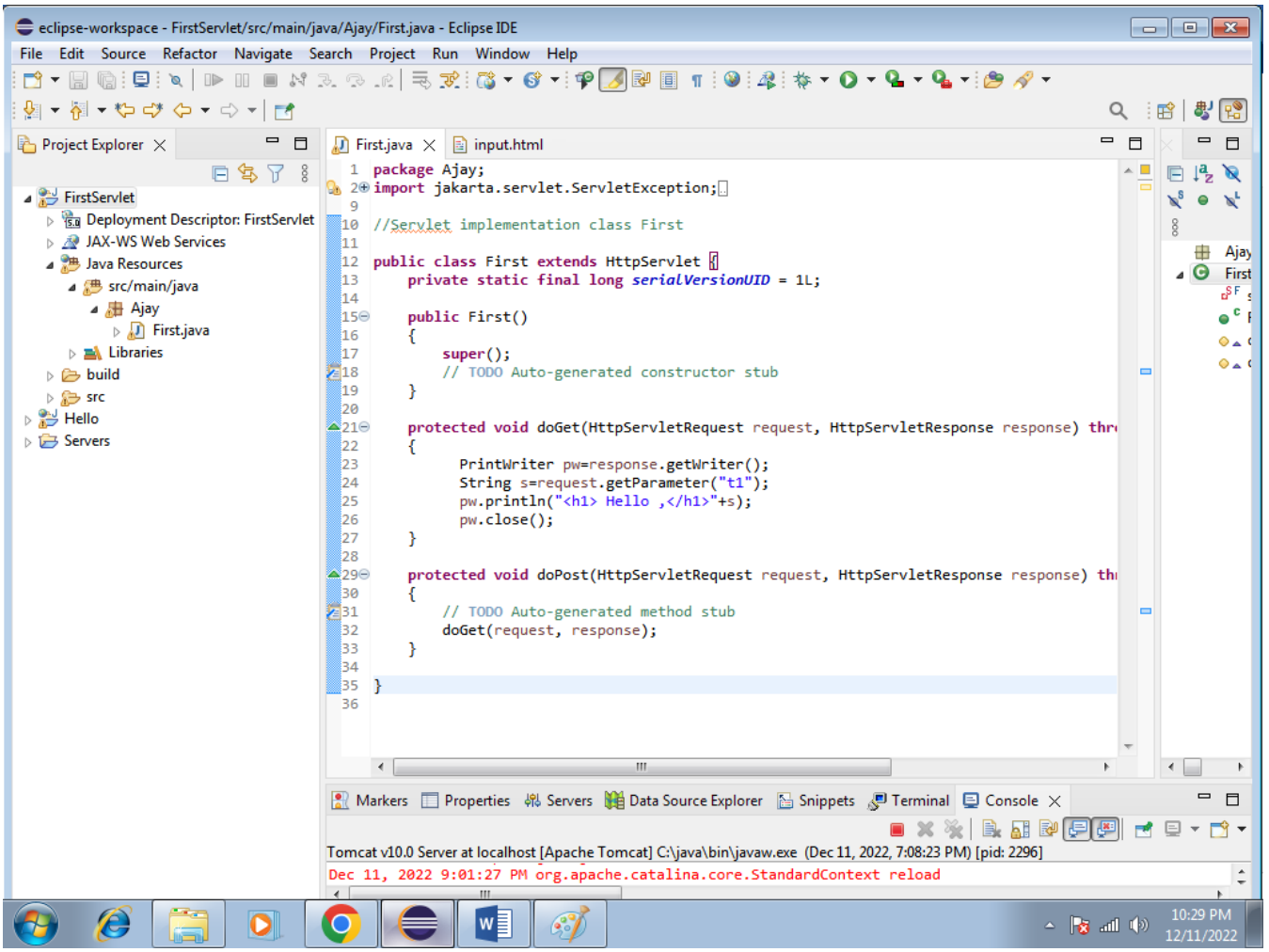
Step 2. Give Project name and click Next. Tick the checkbox that says **Generate web.xml deployment descriptor.**

Initial Project structure:
Upon creation of project, the hierarchy (project structure) would look like this:



Step 3. Create a Servlet class:

We are creating a Http Servlet by extending HttpServlet class. Right click on the src folder and create a new class file, name the file as MyServletDemo. The file path should look like this: Java Resources/src/default package/MyServletDemo.java



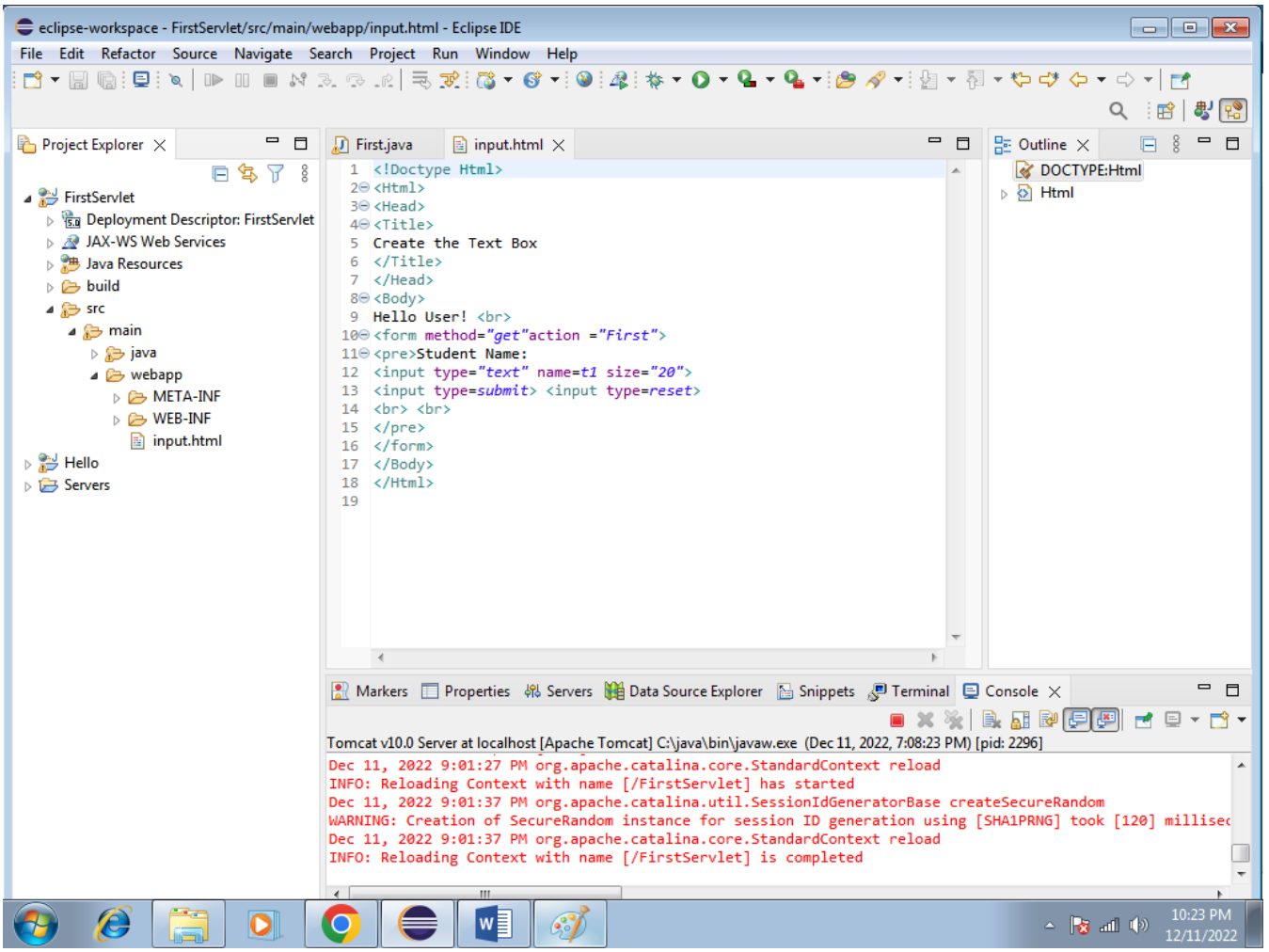
Edit following java code related to our first servlet:

First.java

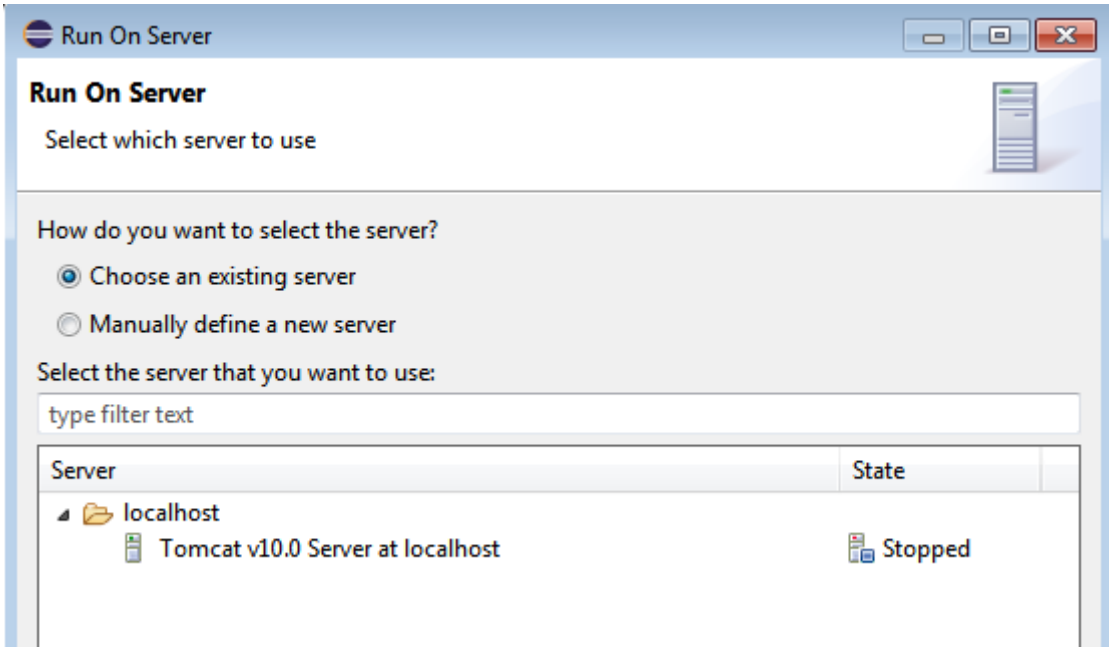
```
package Ajay;
import jakarta.servlet.ServletException;
import jakarta.servlet.annotation.WebServlet;
import jakarta.servlet.http.HttpServlet;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;

//Servlet implementation class First

public class First extends HttpServlet {
        private static final long serialVersionUID = 1L;

    public First()
    {
        super();
        // TODO Auto-generated constructor stub
```

```
    }

        protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException
        {
                PrintWriter pw=response.getWriter();
                String s=request.getParameter("t1");
                pw.println("<h1> Hello ,</h1>"+s);
                pw.close();
        }

protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException
        {
                // TODO Auto-generated method stub
                doGet(request, response);
        }

}
```

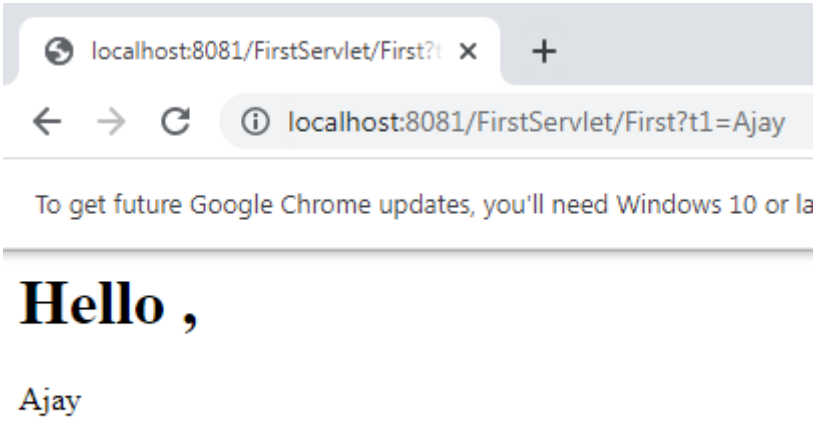
Step 4: Create an html page to call the servlet class on a webpage
We are creating an html file that would call the servlet once we click on the link on web page. Create this file in src/main/webapp folder. The path of the file should look like this:
src/main/webapp/input.html



Step 5: **Run the project:**
Right click on the input.html, select Run As option then select Run on Server. In next window select the following option i.e Tomcat v 10.0 Server at localhost.



Then simply click on Next and Finish button to see the following output:



W7 D1

What is Maven?

Maven is a powerful project management tool that is based on POM (project object model). It is used for projects build, dependency and documentation. It simplifies the build process like ANT. But it is too much advanced than ANT.

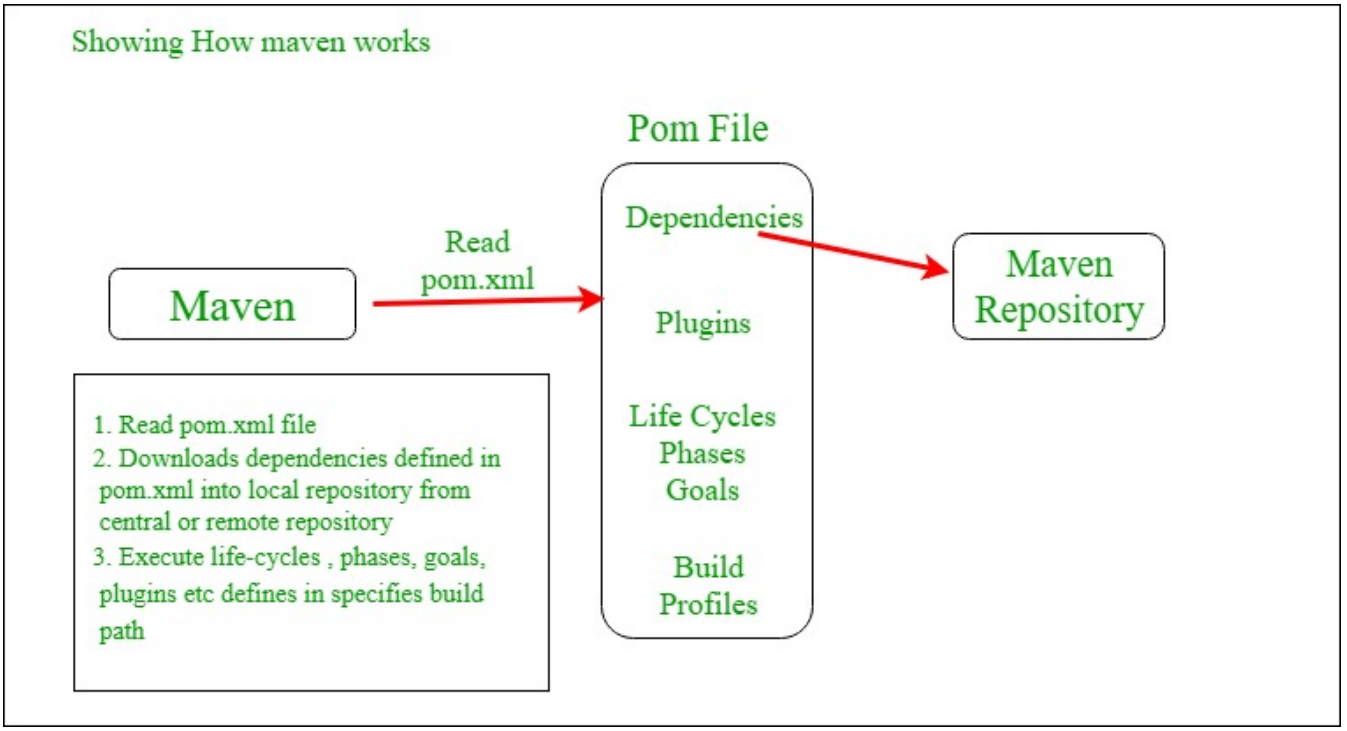
In short terms we can tell maven is a tool that can be used for building and managing any Java-based project. maven make the day-to-day work of Java developers easier and generally help with the comprehension of any Java-based project.

What maven does?

Maven does a lot of helpful task like

1. We can easily build a project using maven.
2. We can add jars and other dependencies of the project easily using the help of maven.
3. Maven provides project information (log document, dependency list, unit test reports etc.)
4. Maven is very helpful for a project while updating central repository of JARs and other dependencies.
5. With the help of Maven we can build any number of projects into output types like the JAR, WAR etc without doing any scripting.
6. Using maven we can easily integrate our project with source control system (such as Subversion or Git).

How maven works?



Core Concepts of Maven:

1. **POM Files:** Project Object Model(POM) Files are XML file that contains information related to the project and configuration information such as dependencies, source directory, plugin, goals etc. used by Maven to build the project. When you should execute a maven command you give maven a POM file to execute the commands. Maven reads pom.xml file to accomplish its configuration and operations.
2. **Dependencies and Repositories:** Dependencies are external Java libraries required for Project and repositories are directories of packaged JAR files. The local repository is just a directory on your machine hard drive. If the dependencies are not found in the local Maven repository, Maven downloads them from a central Maven repository and puts them in your local repository.
3. **Build Life Cycles, Phases and Goals:** A build life cycle consists of a sequence of build phases, and each build phase consists of a sequence of goals. Maven command is the name of a build lifecycle, phase or goal. If a lifecycle is requested executed by giving maven command, all build phases in that life cycle are executed also. If a build phase is requested executed, all build phases before it in the defined sequence are executed too.
4. **Build Profiles:** Build profiles a set of configuration values which allows you to build your project using different configurations. For example, you may need to build your project for your local computer, for development and test. To enable different builds you can add different build profiles to your POM files using its profiles elements and are triggered in the variety of ways.
5. **Build Plugins:** Build plugins are used to perform specific goal. you can add a plugin to the POM file. Maven has some standard plugins you can use, and you can also implement your own in Java.

Installation process of Maven

The installation of Maven includes following Steps:

1. Verify that your system has java installed or not. if not then install java (Link for Java Installation)
2. Check java Environmental variable is set or not. if not then set java environmental variable.(link to install java and setting environmental variable)
3. Download maven (Link)
4. Unpack your maven zip at any place in your system.
5. Add the bin directory of the created directory apache-maven-3.5.3(it depends upon your installation version) to the PATH environment variable and system variable.
6. open cmd and run **mvn -v** command. If it print following lines of code then installation completed.

```
Apache Maven 3.5.3 (3383c37e1f9e9b3bc3df5050c29c8aff9f295297; 2018-02-25T01:19:05+05:30)
Maven home: C:\apache-maven-3.5.3\bin\..
Java version: 1.8.0_151, vendor: Oracle Corporation
Java home: C:\Program Files\Java\jdk1.8.0_151\jre
Default locale: en_US, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"
```

Maven pom.xml file

POM means Project Object Model is key to operate Maven. Maven reads pom.xml file to accomplish its configuration and operations. It is an XML file that contains information related to the project and configuration information such as **dependencies**, **source directory**, **plugin**, **goals etc**. used by Maven to build the project.

The sample of pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">

      <modelVersion>4.0.0</modelVersion>
      <groupId> com.project.loggerapi </groupId>
      <artifactId>LoggerApi</artifactId>
      <version>0.0.1-SNAPSHOT</version>

    <!-- Add typical dependencies for a web application -->
    <dependencies>
        <dependency>
            <groupId>org.apache.logging.log4j</groupId>
            <artifactId>log4j-api</artifactId>
            <version>2.11.0</version>
         </dependency>
    </dependencies>

  </project>
```

Elements used for Creating pom.xml file

1. **project-** It is the root element of the pom.xml file.
2. **modelVersion-** modelversion means what version of the POM model you are using. Use version 4.0.0 for maven 2 and maven 3.
3. **groupId-** groupId means the id for the project group. It is unique and Most often you will use a group ID which is similar to the root Java package name of the project like we used the groupId com.project.loggerapi.
4. **artifactId-** artifactId used to give name of the project you are building.in our example name of our project is LoggerApi.
5. **version-** version element contains the version number of the project. If your project has been released in different versions then it is useful to give version of your project.

Other Elements of Pom.xml file

1. **dependencies-** dependencies element is used to defines a list of dependency of project.
2. **dependency-** dependency defines a dependency and used inside dependencies tag. Each dependency is described by its groupId, artifactId and version.
3. **name-** this element is used to give name to our maven project.
4. **scope-** this element used to define scope for this maven project that can be compile, runtime, test, provided system etc.
5. **packaging-** packaging element is used to packaging our project to output types like JAR, WAR etc.

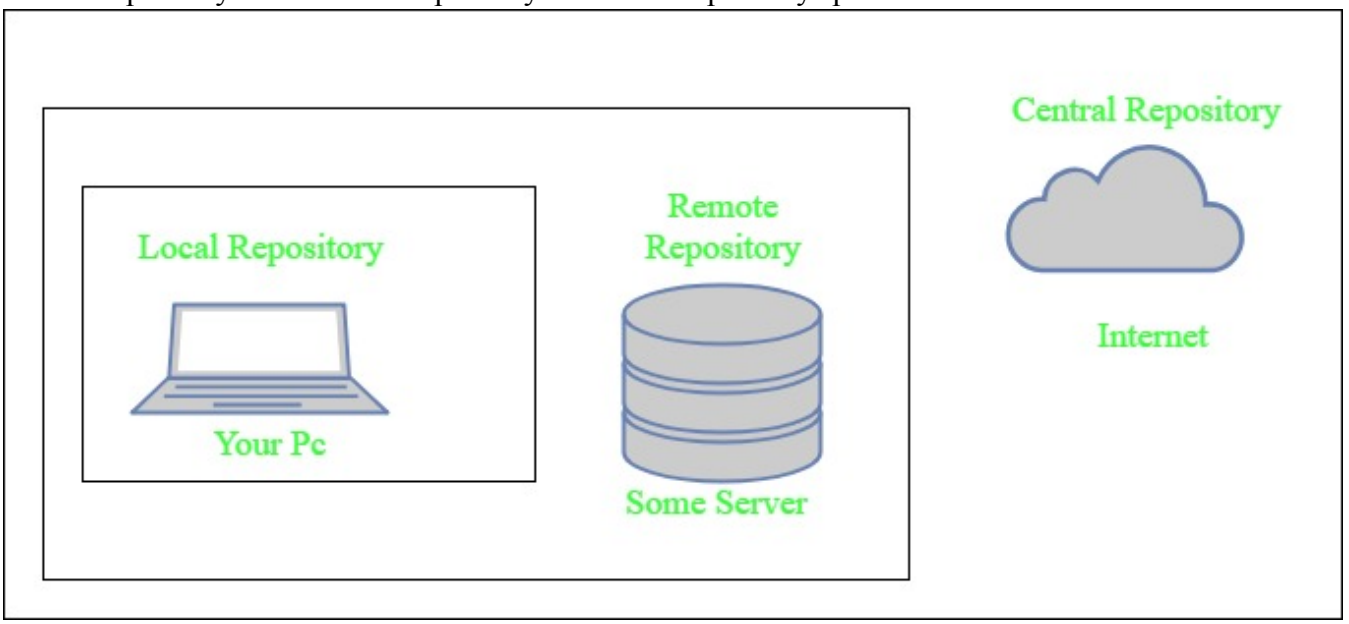
Maven Repository

Maven repositories are directories of packaged JAR files with some metadata. The metadata are POM files related to the projects each packaged JAR file belongs to, including what external dependencies each packaged JAR has. This metadata enables Maven to download dependencies of your dependencies recursively until all dependencies are download and put into your local machine.

Maven has three types of repository :

1. Local repository
2. Central repository
3. Remote repository

Maven searches for dependencies in this repositories. First maven searches in Local repository then Central repository then Remote repository if Remote repository specified in the POM.



1. **Local repository-** A local repository is a directory on the machine of developer. This repository contains all the dependencies Maven downloads. Maven only needs to download the dependencies once, even if multiple projects depends on them (e.g. ODBC).
   By default, maven local repository is user_home/m2 directory.
   example – **C:\Users\asingh\.m2**
2. **Central repository-** The central Maven repository is created Maven community. Maven looks in this central repository for any dependencies needed but not found in your local repository. Maven then downloads these dependencies into your local repository. You can view central repository by this link.
3. **Remote repository-** remote repository is a repository on a web server from which Maven can download dependencies.it often used for hosting projects internal to organization. Maven then downloads these dependencies into your local repository.

Practical Application Of Maven

When working on a java project and that project contains a lot of dependencies, builds, requirement, then handling all those things manually is very difficult and tiresome. Thus using some tool which can do these works is very helpful.

Maven is such a build management tool which can do all the things like adding dependencies, managing the classpath to project, generating war and jar file automatically and many other things.

Pros and Cons of using Maven

Pros:

1. Maven can add all the dependencies required for the project automatically by reading pom file.
2. One can easily build their project to jar, war etc. as per their requirements using Maven.

3. Maven makes easy to start project in different environments and one doesn't needs to handle the dependencies injection, builds, processing, etc.
4. Adding a new dependency is very easy. One has to just write the dependency code in pom file.

Cons:

5. Maven needs the maven installation in the system for working and maven plugin for the ide.
6. If the maven code for an existing dependency is not available, then one cannot add that dependency using maven.

When should someone use Maven?

One can use the Maven Build Tool in the following condition:

1.When there are a lot of dependencies for the project. Then it is easy to handle those dependencies using maven.

2.When dependency version update frequently. Then one has to only update version ID in pom file to update dependencies.

3.Continuous builds, integration, and testing can be easily handled by using maven.

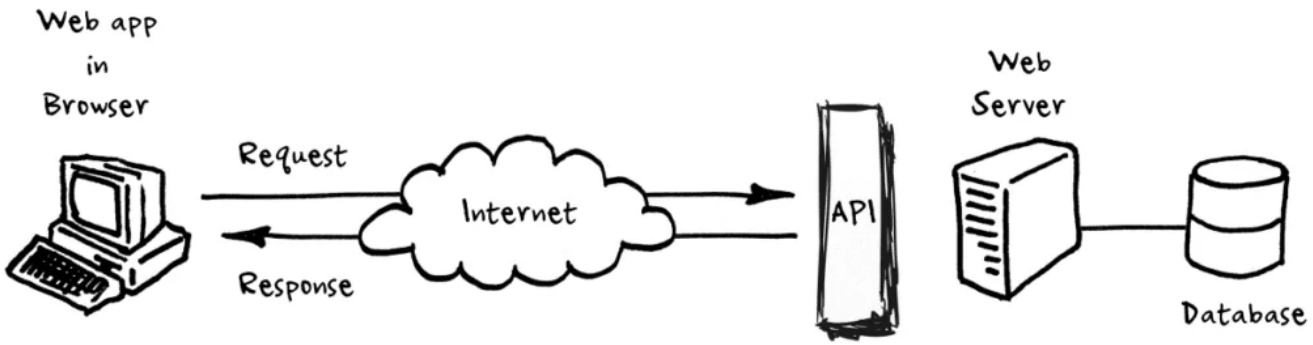
4.When one needs an easy way to Generating documentation from the source code, Compiling source code, Packaging compiled code into JAR files or ZIP files.

W8 D4

Application Programming Interface (API)

API, which is the acronym for Application Programming Interface, is a series of rules and protocols that helps applications fetch data and data points from other systems. An effective API will make the integration of two computer systems easily communicate to each other, as it will have all necessary building blocks put together by the computer programmer.

If we look at the below diagram, which is a basic information flow diagram depicting the position of an Application Programming Interface in the whole schema of system integration.



In simple APIs are also known as connectors that establish a synchronized connection between two applications for the exchange of information on a real-time basis.

How does an API work?

Let us understand the working of an Application Programming Interface with a very simple example. We all are fond of watching movies in the cinema. When we plan to book a ticket, we go to the booking website, make our selection, enter credit card details, and the tickets are ready, and we print it out.

However, what seems very simple on the front is a series of information exchanges between the website that you see on the front and the database at the back. How do you think this happens? Well, it is the Application Programming Interface that connects the website and the database, and this integration happens in real-time, which is why you feel an instant transaction, within seconds.

This was an example of a B2C (business to customer) transaction. However, the same is applicable in B2B transactions too. The only difference is that in this case, the request command comes from one business application, and then the information is fetched from another business system to complete the transaction.

The fact that this data flow between two business applications has become the backbone of today's enterprises. In the case of B2B integrations, APIs have an important role to transmit the data.

Why do we need API?

The importance of the Application Programming Interface is multifold. The role of API is to allow usage of data to the users in a faster, easier, and efficient way, for a specific purpose.

We have seen Application Programming Interface been used significantly in the private sector across enterprises, globally. However, it has been observed that governments are also now using APIs across their systems because it makes their applications agile and flexible.

While we agree that legacy systems have been operating in silos and today enterprises who want to transition from legacy systems to the latest cloud-based systems, APIs form an integral part of this whole framework.

Furthermore, organizations that are already on the path to digital transformation and have implemented different systems for different objectives, are looking to interconnect the applications so that the data is centrally available. This is another reason why APIs are needed as they will eventually help organizations to make informed decisions.

While companies have concerns on the security aspects of Application Programming Interface, but today programmers ensure that the building blocks of these APIs are strong. Moreover, APIs come with a series of permissions and audit trails related to the security of the application.

Technology is continuously evolving and Application Programming Interface are going to play a key role in the new dimension to the way applications will operate.

Examples of APIs

To understand what APIs do and how they operate, here are few examples that will help you get a deeper insight into APIs.

1. **Weather Data**

One common example of API that we see on a day-to-day basis is the data on weather. Our smartphones have this weather app that shares snippets on the weather.

2. **Login Authentication**

   We see many websites and applications, which prompt users to log in using social media accounts such as Facebook, Twitter, LinkedIn, and Google.

3. **Payment**

   When we order from e-commerce sites like Amazon, we get multiple options to make payments such as PayPal. Here the Amazon fetches information from the PayPal account and executes the payment.

4. **Travel booking**

   On travel booking sites, we see that the site fetches information about flight schedules and hotel availabilities instantly. This happens through APIs where the booking portal interacts with the respective airline website or hotel website

These are just a few examples to give you an idea of the usability of APIs. However, there are many more examples in B2B and B2C space.

Types of APIs

There are many forms of Application Programming Interface. Software developers can use a variety of protocols and rules to build an API. Here we will look at some of the API types:

- **Web APIs**

  These are connectors that use the HTTP protocol. Web Application Programming Interface is used to interact with the browser, which will have web notifications and storage on the web. If we combine multiple Web APIs, then we can create a composite API, which is a collection of data or service Application Programming Interface.

- **Open APIs**

  These are also known as external APIs or public APIs. These Application Programming Interface are easily available to developers without any restrictions. Occasionally there might be some registration to be done to get an API key or it may be completely open.

- **Partner APIs**

  Technically these are similar to open Application Programming Interface, however, there is a restriction to access these. The access is usually controlled by a third-party API gateway.

- **Composite APIs**

  These are a combination of different Application Programming Interface that helps developers to access multiple data touchpoints in a single call. These touchpoints can be fetched through a single API or it can have multiple APIs fetching information from multiple data sources.

When it comes to successfully developing an Application Programming Interface, there are certain rules that developers need to follow. These are known as API protocols. One of the most common protocols that any developer follows is the REST API protocol.

REST known as Representational State Transfer is a very commonly used and popular Web Application Programming Interface framework. There are certain fundamentals around architecture and principles that developers need to adhere to.

The other type of Application Programming Interface protocol that is used by developers is the JSON-RPC and XML-RPC protocol. In both these protocols, an API call can contain multiple parameters.

Benefits of using APIs

For organizations, APIs are a way of developing applications that can deliver a better customer experience. Some of the key benefits of using APIs are as follows:

- **Easy to Integrate**

  APIs are connectors that help in connecting two applications and systems for data flow. This will ensure that the overall user experience is seamless.

- **Improved Integration**

  Having APIs with the right set of protocols helps in creating a better-integrated solution that can help organizations to manage their data and processes, thereby helping them gain better visibility.

- **Automation**

  While applications are being integrated using APIs, it automates various business processes and the tasks associated with them. This helps in bringing about great efficiency among various functions.

- **Enhanced Services**

  Having APIs makes the implementation of multiple systems very easy. These help in the case of systems, involving third-party applications as well.

- **Innovation**

  APIs play a critical role in the whole process of digital transformation for any organization. They help in developing unique business models by harnessing the power of technology.

Types of API Protocols (SOAP, REST)

As we know that each machine understand and deals in its different language or input so web-services are something which are required for inter communication between machines and to exchange data between them. In order to implement some set of restrictions over their communication some set of rules and regulations are defined which are known as

web-services which basically defines the format and type of data that need to be exchanged and specifically a contract which both machines should be aware of before taking part in communication.

This communication system can be categorized into two types, namely Simple Object Access Protocol or SOAP, and Representational State Transfer or REST.

Following are the important differences between REST API and SOAP API.

S.N	Key	REST API	SOAP API
1	Implementation	Rest API is implemented as it has no official standard at all because it is an architectural style.	On other hand SOAP API has an official standard because it is a protocol.
2	Internal communication	REST APIs uses multiple standards like HTTP, JSON, URL, and XML for data communication and transfer.	SOAP APIs is largely based and uses only HTTP and XML.
3	Resource requirement	As REST API deploys and uses multiple standards as stated above, so it takes fewer resources and bandwidth as compared to SOAP API.	On other hand Soap API requires more resource and bandwidth as it needs to convert the data in XML which increases its payload and results in the large sized file.
4	Description	REST API uses Web Application Description Language for describing the functionalities being offered by web services.	On other hand SOAP API used Web Services Description language for the same.
5	Security	REST has SSL and HTTPS for security.	On other hand SOAP has SSL(Secure Socket Layer) and WS-security due to which in the cases like Bank Account Password, Card Number, etc. SOAP is preferred over REST.
6	Abbreviation	REST stands for Representational State Transfer.	On other hand SOAP stands for Simple Object Access Protocol
7	Interchange	REST can make use of SOAP as the underlying protocol for web services, because in the end it is just an architectural pattern.	On other hand SOAP cannot make use of REST since SOAP is a protocol and REST is an architectural pattern.

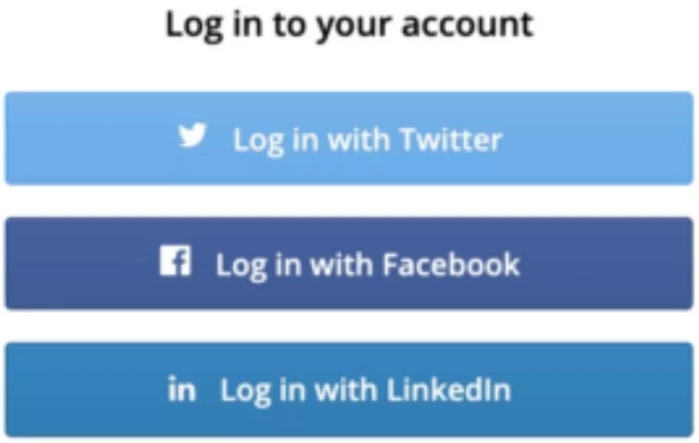
Examples of APIs Used in Our Daily Lives

It is not true if you think you're far away from the APIs, as we all use different APIs in our daily lives. Some of the top APIs used daily include:

1. Social Media Bots

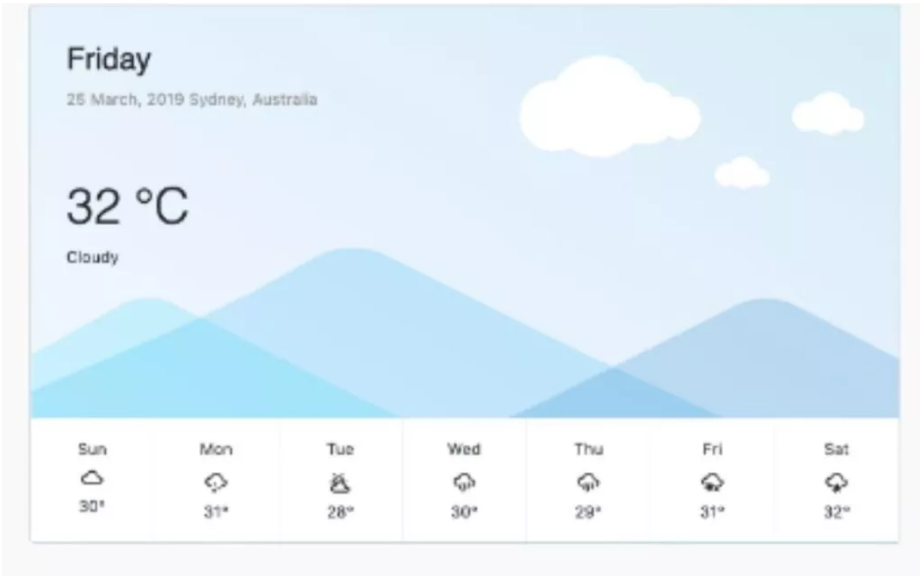
How much time do you spend on social media platforms like Twitter? All leading social media platforms use different bots which use social media APIs for different tasks. The social media bots leverage the social media APIs to integrate successfully into the social media platform's internal software system. Some social media bots include generating great stories seamlessly, tagging bots under different threads, etc. It can wrap all text from different threads and present it as readable text.

2. **Log-In Using XYZ**



Have you seen these options on online platforms like "Log-in using Google" or "Log-in using Facebook?" These options allow users to log in to the online services without opening a separate account on the platform. APIs are used to connect the different platforms while logging- in using different accounts. The addition of login details on different online platforms must integrate well with the XYZ platforms.

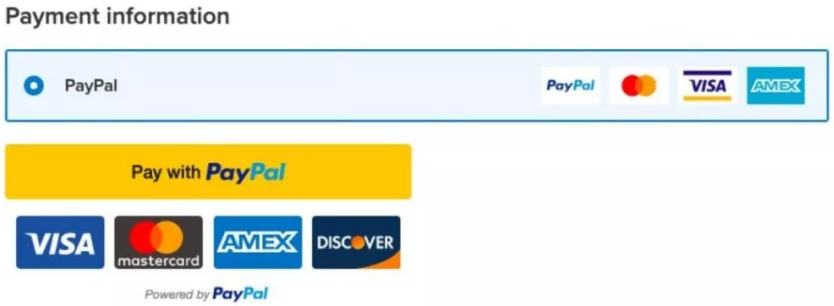
3.**Weather Snippets**



Do you see weather data on your smartphones? Weather data is one of the common forms of APIs. These weather snippets are on different platforms like smart home devices, Apple's Weather app, Google Search, etc. The search results in the weather snippets yield current weather conditions and

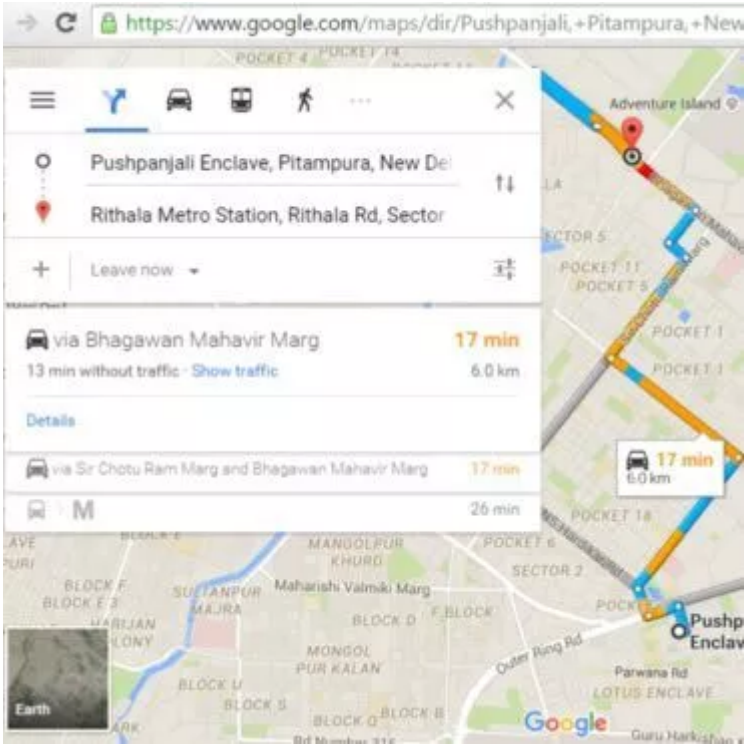
forecasts. This information is sourced from the third-party application by the different platforms. These third parties share the latest weather details, which are easy to reformat according to the platform. Different weather APIs cover weather snippets.

4. **Pay with PayPal**



How soon do you pay with PayPal monthly? PayPal is one of the widely used examples of APIs which allows users to connect personal financial information to their PayPal accounts. It is a leading fintech service which offers easy and secure money transfers while staying embedded in different websites requiring online transactions. These websites don't have direct access to bank or card information which is ensured with the dedicated API integrations only.

5**. Google Maps**



Who doesn't use Google Maps? The Google Maps API allows several geographic aptitudes to users within seconds. It is easy to locate places like relative distance from the current location, reviews,

business hours, contact information, restaurants, etc. Google Maps application leads to the goes to the Google Maps website to offer quick information.

W8 D2

API endpoints :

What Is An API Endpoint?

An API endpoint is a digital location where an API receives requests about a specific resource on its server. In APIs, an endpoint is typically a uniform resource locator (URL) that provides the location of a resource on the server.

To fully understand this definition and where endpoints fit in the API model, let's briefly review how APIs work.

For two software applications to integrate over the internet, one application — called the client — sends a request to the other application's API. The client may request a resource from the app's database or ask to perform an action on the server.

Upon receiving and validating the client's request, the API performs the requested action, then sends a response back to the client. This response includes the status of the request (e.g., completed or denied) and any resources requested by the client.

APIs typically allow access to many different resources on a server. For example, a social network's API might let clients retrieve and modify post content, user profiles, and images. A news site's API will allow access to its article content, authors, and media like podcasts and videos.

Knowing this, how do clients specify which resource they want to access in their request?

The answer is by using the correct endpoint. In their requests, clients specify an endpoint as a URL. This URL tells the server, "The resource I want is at this location."

The process is similar to how you access web pages in a browser. Web browsers load web pages by sending a URL to a web server, and the server responds with the requested page. Similarly, the client needs the right endpoint URL to request a particular resource from an API.

Endpoint vs. API

It's important to note that endpoints and APIs are different. An endpoint is a component of an API, while an API is a set of rules that allow two applications to share resources. Endpoints are the locations of the resources, and the API uses endpoint URLs to retrieve the requested resources.

API Endpoint Examples

Developers list all endpoints of their API in the documentation so that users know how to access the resources they need. But what do these endpoints look like in practice? Let's see some real-world examples from leading platforms.

1.Twitter API Endpoint Example

The Twitter API exposes data about tweets, direct messages, users, and more.

Let's say you want to retrieve the content of a specific tweet. To do this, you can use the tweet lookup endpoint, which has the URL **https://api.twitter.com/2/tweets/{id}** (where **{id}** is the unique identifier of the tweet).

Now, say you want your website to stream public tweets in real-time so your visitors stay informed on a specific topic. You can use Twitter's filtered stream endpoint, whose URL is **https://api.twitter.com/2/tweets/search/stream**.

2.Spotify API Endpoint Example

Spotify's API gives developers access to song, artist, playlist, and user data. For example, if you want to get a specific album, you can access any album in the Spotify catalog with the endpoint **https://api.spotify.com/v1/albums/{id}** (where **{id}** is the album's unique identifier).

Or, say you want to send a request that makes a user follow a playlist. In this case, send a PUT request with the endpoint **https://api.spotify.com/v1/playlists/{playlist_id}/followers** (where **{playlist_id}** is the unique identifier of the playlist).

3.YouTube API Endpoint Example

The YouTube API, among other things, makes it easy to embed YouTube videos on any website. When you go to a YouTube video and copy the embed code, you are requesting the video from YouTube's API.

Another way to get videos through YouTube's API is by requesting them from the endpoint **https://www.googleapis.com/youtube/v3/videos**, which returns a list of videos that match the parameters you specified in your request.

Why are API endpoints important?

One of the first questions you ask about APIs is: Why do so many businesses share their data openly, for free?

Most often, the answer is scale. As companies grow, the staff within those companies realize they have more ideas than they have the time and resources to develop them. It's typically easier to work with other external companies that specialize in these ideas than build them in-house.

By creating APIs, a company allows third-party developers to build applications that improve adoption and usage of its platform. That way, a business can build an ecosystem that becomes dependent on the data from their API, which often leads to additional revenue opportunities.

Take HubSpot APIs as another example: By exposing our software's functionality and data through APIs, developers can integrate their applications with our technology, resulting in an app ecosystem that increases our reach and makes things much easier for customers who wish to use other applications alongside HubSpot.

As we've learned, endpoints are central to APIs, literally. They're the point at which the client and the server communicate. Without properly structured and functioning endpoints, an API will be confusing at best and broken at worst. As you make more data available through your API, it's vital to ensure that each endpoint provides valuable resources for clients.

How to Test API Endpoints

When discussing web APIs, we're usually talking about a type of API called a REST API, which utilizes HTTP methods that tell the API what action to take. The four most common HTTP methods in API requests are:

- **GET**: retrieves a resource
- **POST**: creates a resource
- **PUT**: updates an existing resource
- **DELETE**: removes a resource

Let's see how to make a request with the Twitter API's filtered stream endpoint. Requests are formatted by writing the HTTP method, followed by the endpoint URL. So, a request to the filtered stream endpoint will look like: **POST https://api.twitter.com/2/tweets/search/stream**.

Now, imagine you want to stay informed about HubSpot on Twitter. You'll want to get tweets from the @HubSpot account as soon as they're posted. Also, let's say you only want tweets that contain links to articles.

In that case, using the filtered stream endpoint is a perfect choice. But, in order for the API to know which tweets to send you, you'll need to define filtering criteria. Otherwise, you'll just be asking to see every tweet posted in real-time (gulp).

We can apply filtering criteria to the endpoint in the form of rules. To build these rules, you'll use a set of operators. For this example, you can use two Twitter API operators — **from:** and **has:links** — to only see tweets from certain accounts that contain links. To instruct the filtered stream endpoint to only show tweets from the accounts @HubSpot that contain links, use the following rule: **from:HubSpot has:links**.

In your request, you'll use the HTTP method **POST**. In addition to including the rule mentioned above in your request, you'll include the content type and authorization. Below the content type is defined as "application/json" so the request is rendered in the data format JavaScript Object Notation (JSON).

There are several online tools available for testing an API endpoint. Here we'll use cURL, a command-line tool that supports HTTP. It can make requests, get data, and send data, so it's a great tool for testing APIs.

Here's what your request to the Twitter API should look like on the command line. To authenticate your request, you'll have to replace the placeholder text **$BEARER_TOKEN** with your app's unique bearer token which is available in your developer portal.

```
curl -X POST 'https://api.twitter.com/2/tweets/search/stream/rules' \
-H "Content-type: application/json" \
-H "Authorization: Bearer $BEARER_TOKEN" -d \
'{
"add": [
{"value": "from:HubSpot has:links"}
]
}'
```