Full Stack Developer

Intro to Cloud Computing

**Summary and Highlights**

In this Module 1, you have learned the following:

* Cloud computing delivers on-demand computing resources over the internet on a pay-as-you-go basis; resources are dynamically assigned and reassigned among multiple users and scale up and down in response to users’ needs.
* The origins of cloud computing can be traced back to the mainframes of the 1950s, with virtualization technologies and hypervisors serving as catalysts for the emergence of modern-day cloud computing.
* Organizations must consider their business needs, investment viability, and risk capacity to create a cloud adoption strategy that delivers desired benefits without causing business disruptions and security, compliance, or performance issues.
* Cloud adoption is growing faster than predicted. Driving this technological wave are cloud service providers with a host of services ranging from Infrastructure, Platform, and Software services. Some major Cloud providers of our time include AWS, Alibaba Cloud, Google, IBM, and Microsoft Azure.
* The adoption of cloud technologies enables enterprises, big and small, to be agile, innovative, and competitive and to create differentiated customer experiences. Organizations are asking not whether they should move to the cloud but rather what strategy they should adopt to move to the cloud.
* Some case studies that demonstrate the impact businesses have created by adopting the cloud:
* American Airlines is adopting cloud technologies to deliver customer value rapidly across its enterprise
* UBank leveraging cloud platform services to give more control to their developers, thereby removing barriers to innovation
* Bitly leveraging the scalability offered by cloud infrastructure for low-latency delivery to its geographically dispersed enterprise customers
* ActivTrades leverages the infrastructure, storage, network, and security offerings on the cloud to accelerate the execution and delivery of new functions in their online trading systems to their customers
* Emerging technologies powered by the cloud are disrupting existing business models and creating unprecedented opportunities for businesses to grow, innovate, and create value for their customers.
* Some case studies that demonstrate how the use of emerging technologies on the cloud is creating value for millions around the world:
* The use of the Internet of Things on the Cloud to combat poaching of endangered rhinos in South Africa
* Artificial Intelligence on the cloud is being leveraged to deliver unique digital experiences to millions of fans around the world by the United States Tennis Association
* Blockchain on the cloud helps farmers reduce waste by building traceability and transparency in the food supply chain
* The use of data analytics for driving predictive maintenance solutions for a city’s infrastructure by KONE

**Summary and Highlights**

In this module 2, you have learned:

* With IaaS, the cloud provider manages physical resources.
* With PaaS, the provider manages the platform infrastructure.
* In the SaaS model, the provider hosts and manages the applications and data.
* Infrastructure-as-a-Service is a form of cloud computing that delivers fundamental computer, network, and storage resources to consumers on-demand, over the network, on a pay-as-you-go basis.
* The key components of cloud infrastructure are:
  + Physical data centers
  + Compute
  + Network
  + Storage
* Platform-as-a-Service is a cloud computing model that provides customers with a complete platform—hardware, software, and infrastructure.
* The high level of abstraction, support services, runtime environments, rapid deployment mechanisms, and middleware capabilities distinguishes PaaS clouds.
* PaaS advantages are:
  + Scalability
  + Faster time to market products and services.
  + Greater agility and innovation
* Software-as-a-Service is a cloud offering that provides users with access to a service provider’s cloud-based software.
* SaaS characteristics are:
  + Multitenant architecture
  + Security, compliance, and maintenance
  + Customization of applications
  + Subscription model
  + Scaling
* SaaS advantages are:
  + Direct procurement of solutions
  + Improved workforce productivity and efficiency
  + Enable distribution of software costs
* Deployment models indicate where the infrastructure resides, who owns and manages it, and how cloud resources and services are made available to users. There are four main deployment models available on the cloud—public, private, hybrid, and community.
* In the public cloud model, the service provider owns, manages, provisions, and maintains the physical infrastructure such as data centers, servers, networking equipment, and storage, with users accessing virtualized computing, networking and storage resources as services.
* In the private cloud model, the provider provisions the cloud infrastructure for exclusive use by a single organization. The private cloud infrastructure can be internal to the organization and run or on-premises. Or it can be on a public cloud, as in the case of Virtual Private Clouds (VPC), and be owned, managed, and operated by the cloud provider.
* In the hybrid cloud model, an organization’s on-premise private cloud and a third-party, public cloud are connected as a single, flexible infrastructure that leverages the features and benefits of both Public and Private clouds.
* In the community cloud model, the provider provisions the cloud infrastructure for use by a community of organizations with shared concerns. One or more of the organizations in the community, a third-party provider, or both are responsible for the ownership, management, and operation of this infrastructure.

**Summary and Highlights**

**In this module 3, you have learned:**

* Cloud infrastructure consists of data centers, storage, networking components, and computing resources.
* Virtualization is the process of creating a software-based version of physical resources, made possible by hypervisors.
* A few different types of Virtual Machines can be provisioned on the cloud. These include:
  + Shared or Public Cloud VMs are provider-managed, multi-tenant deployments that can be provisioned on-demand with predefined sizes
  + Transient or Spot VMs that take advantage of unused capacity in a cloud data center
  + Reserved VMs that allow you to reserve capacity and guarantee resources for future deployments
  + Dedicated hosts that offer single-tenant isolation
* Bare metal servers are single-tenant physical servers that are dedicated to a single customer. Bare metal servers fulfill the demanding needs of high-performance computing (HPC) and data-intense applications. They are ideal for applications that have a high degree of security or compliance requirements.
* Networking capabilities in the cloud are delivered as a service rather than in the form of rack-mounted devices. Cloud resources such as VMs (or VSIs), storage, network connectivity, and load balancers are deployed into subnets within Virtual Private Clouds (VPCs). Using private and public subnets allows users to deploy multi-tier enterprise applications securely. Load balancers distribute the traffic and allow applications to be responsive.
* Containers are executable units of software in which application code, its libraries, and its dependencies are packaged in a common way, so that it can be run anywhere, from desktops to traditional IT, to the cloud. Containers are more lightweight and consume fewer resources than Virtual Machines, helping streamline the development and deployment of cloud-native applications.
* Cloud storage is available in four main types–Direct Attached, File, Block, and Object Storage. These storage types differ in how they can be accessed, the capacity they offer, how much they cost, the types of data they are best suited to store, and their read-write speed.
* Direct Attached (or Local) Storage is storage that is presented directly to a cloud-based server and is effectively either within the host server chassis or within the same rack.
* File Storage is typically presented to compute nodes as a Network File System (NFS), which means that the storage is connected to compute nodes over a standard ethernet network.
* Block Storage is presented to compute nodes using high-speed fiber connections, typically provisioned in volumes, which are mounted onto a compute node.
* Object Storage is accessed via an API and doesn’t need an underlying compute node.
* Object Storage offers infinite capacity as you can keep adding files to it and just pay for what you use. Compared to the other storage types, object storage is the slowest in terms of read and write speeds.
* A Content Delivery Network (CDN) is a distributed server network that accelerates internet content delivery by delivering temporarily stored or cached copies of website or media content to users based on their geographic location.

**SUMMARY AND HIGHLIGHTS**

In this module 4, you have learned:

* Hybrid multi-cloud is a cloud adoption strategy that makes it possible for public clouds, private clouds, and on-premises IT to interoperate seamlessly while leveraging the best cloud-based services from different public cloud providers.
* Microservices architecture is an approach in which an application is built as a collection of loosely coupled and independently deployable components or services, leading to efficient development, maintenance, and upgradation cycles.
* Serverless computing is an approach to computing that offloads responsibility for common infrastructure management tasks for application runtimes to cloud providers, allowing developers to focus their time and effort on development and testing, and not have to worry about provisioning, maintaining, and scaling compute resources.
* Cloud native applications are applications that are built or refactored to work in the cloud environment. These applications, developed using DevOps methodologies, consist of microservices packaged in containers that can run in any environment—making it possible to create and update features in quick iterative cycles.
* DevOps is a collaborative approach that enables development and operations teams to continuously deliver software in quick iterative cycles while reducing overhead, duplication, and rework. DevOps’ tools, practices, and processes help tackle the complexities and challenges posed by the cloud, allowing solutions to be delivered and updated quickly and reliably.
* Application modernization helps organizations accelerate their digital transformation, take advantage of new technologies and services, and become more responsive to changing market dynamics. Cloud computing is one of the key enablers of application modernization.

**Summary and Highlights**

In this module 5, you have learned:

* Cloud security refers to the policies, technological procedures, services, and solutions designed to secure enterprise applications and data on the cloud against insider threats, data breaches, compliance issues, and organized security threats.
* Cloud security is a shared responsibility between the cloud provider and the user organization.
* Security architecture and methods for achieving continuous security need to be embedded through the life cycle of an application to ensure that the application runs on a safe platform, the code is free from vulnerabilities, and the operational risks are understood.
* Identity and Access Management, also known as access control, helps authenticate and authorize users and provides user-specific access to cloud resources, services, and applications.
* As part of their Identity and Access Management services, most cloud providers offer users the ability to define access groups and create access policies that define permissions for users on account resources.
* Cloud encryption, often called the last line of defense, encrypts data and provides robust data access control, key management, and certificate management.
* Data needs encryption in three states:
  + Encryption at rest: Protecting data while it is stored
  + Encryption in transit: Protecting data while it is transmitted from one location to another
  + Encryption in us: Protecting data when it is in use in memory
* All connected systems and cloud-based services should be monitored to maintain visibility of all data exchanges between public, private, and hybrid cloud environments. This ensures that the cloud provides a trusted platform to integrate with your enterprise data centers securely.
* Businesses all over the world are realizing tangible benefits from the use of cloud technologies and services, including:
  + The Weather Company migrating to the cloud to reliably deliver critical weather data at high speed, especially during major weather events such as hurricanes and tornadoes
  + American Airlines uses the cloud platform and technologies to deliver digital self-service tools and customer value more rapidly across its enterprise
  + Cementos Pacasmayo achieving operational excellence and insight to help drive strategic transformation and reach new markets using cloud services
  + Welch chose cloud storage to drive business value from hybrid cloud
  + LiquidPower uses cloud-based SAP applications to fuel business growth
* The market size of the cloud services industry is at nearly three times the growth of overall IT services, increasing the need for qualified cloud computing professionals. Some common job roles that are available in this domain include Cloud Software Engineers, Cloud Integration Specialists, Cloud Data Engineers, Cloud Security Engineers, Cloud DevOps Engineers, and Cloud Solution Architects.

**Introduction to Web Development with HTML5, CSS3, and JavaScript**

In this module 1, you learned that:

* Front-end developers work on the parts of the website or app that the user sees and interact with.
* Back-end developers work on the logic and functionality that keeps the website or app running and responding to users’ inputs.
* Full-stack developers have both sets of skills.
* Front-end developers and back-end developers work closely together.
* Frameworks and libraries extend the functionality of coding languages such as JavaScript and Python.
* Common languages for front-end development include: HTML, CSS, and JavaScript.
* Common languages and frameworks for back-end development include: Python, Django, and Flask.
* Version control systems keep track of changes and resolve conflicts between them.
* CI/CD (Continuous Integration with Continuous Delivery/Deployment) is a best practice developers use to deliver frequent changes reliably.

In this module 2, you learned that:

* HTML provides the basic structure and content for a website using tags.
* Tags represent the elements of an HTML page.
* The HTML DOM Tree describes how a website is structured.
* HTML uses APIs to enhance the user experience, providing features for advanced animation, audio, and video.
* Scripting provides a more interactive user experience when browsing websites.
* It is recommended to not rely on scripting as it can be disabled.
* HTML5 sandboxes help manage iframe mashups.
* HTML5 Browser Support Tables describe which browsers support which HTML5 features.
* JavaScript is used to check if an element is supported by a browser.
* CSS provides consistent style and design throughout the website.
* There are two types of CSS layouts to design websites: fluid and fixed.

In this module 3, you learned that:

* CSS creates a uniform look throughout each element of each page of the website.
* CSS is usually coded in external style sheets and creates base styles for a website.
* CSS frameworks assists in implementing UI elements and creating dynamic web pages.
* CSS has two types of frameworks:
  + Utility-first frameworks, which provide utility classes to help in building one's own styles and layouts.
  + Component frameworks, which provide a wide selection of pre-styled components and templates that can be implemented onto a website.
* Plain (Vanilla) CSS lets developers write the styles and layouts of a website.
* HTML5 elements provide structure and function to websites.
* HTML5 uses the <input> tag to allow users to input information.

In this module 4, you learned that:

* JavaScript is a scripting language that enables developers to add dynamic content to webpages.
* JavaScript variables are declared using the var keyword and take their type from the value assigned.
* Program execution is controlled by statements like If…Then…Else, Switch, For loops, and While loops.
* JavaScript uses blocks of code, called functions, that can be called from anywhere in the script.
* New methods and properties can be added to an object by modifying the prototype for that object.
* Prototypes allow you to define properties and methods for all instances of a specific object.
* Client-side scripts are programs that accompany HTML documents and are used by developers to incorporate more interactive elements.
* The script tag can incorporate a script within an HTML document or call a script from an external file.
* The Document Object Model (DOM) is the programming interface between HTML or XHTML and JavaScript.
* Developers can access HTML DOM elements from JavaScript scripts using the correct DOM notation.
* APIs are often used to access HTML DOM elements in web pages.

**Introduction to Cloud Native, DevOps, Agile, and NoSQL**

Congratulations! You have completed this module 1. At this point in the course, you know:

* A cloud native application consists of microservices working together to comprise an easily scalable application.
* Containers are executable software units that include all of the application code’s microservices, libraries, and dependencies for universal execution.
* The CNCF is an independent body that guides the use and evolution of cloud native technologies.
* The CNCF helps participating companies by providing marketing and support for their projects.
* Hybrid cloud infrastructure is a combination of hosted public and private cloud spaces.
* Application modernization involves building or refactoring legacy applications to enable them to run on the cloud.
* Application modernization helps improve developer productivity and reduce costs.
* TDD, BDD, CI/CD, Agile, Scrum, and SRE are approaches that enable rapid and reliable deployment.

Congratulations! You have completed this module 2. At this point in the course, you know:

* + DevOps is a cultural change in which development and operations engineers work together during the entire development lifecycle.
  + TDD means test cases drive the design and development of code and allows you to develop faster and more confidently.
  + The TDD helps you to develop the logic in the code.
  + BDD helps test the system’s behavior from the outside to ensure that the system behaves as intended.
  + The appropriate levels of performing BDD in software testing are integration, system, and acceptance testing.
  + Continuous Integration is the process of building, testing, and integrating each change into the main branch after the tests pass.
  + Continuous Delivery ensures that code can be deployed to production quickly and safely by deploying every change to a production-like environment.
  + CI/CD gives you faster reaction times to changes and reduces code integration risk.
  + They provide higher code quality because everything is constantly reviewed and tested, and they confirm that the code in version control works.

Go over module and read cover remember

Use linters (XO) with Javascript

Get screenshots of cheat sheats

Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **BPM** | Business Process Management |
| **Composite multicloud** | A variant of hybrid multicloud, distributes single applications across multiple providers, allowing you to move application components across cloud services and vendors as needed |
| **CRM** | Customer Relationship Management |
| **HCM** | Human Capital Management |
| **Hybrid cloud** | A computing environment that connects an organization’s on-premises private cloud and third-party public cloud into a single, flexible infrastructure for running the organization’s applications and workloads |
| **Hybrid monocloud** | A hybrid cloud with one cloud provider |
| **Hybrid multicloud** | An open standards-based stack that can be deployed on any public cloud infrastructure |
| **IaaS** | Infrastructure as a service is a form of cloud computing that delivers fundamental compute, network, and storage resources to consumers on-demand, over the network, on a pay-as-you-go basis |
| **IoT** | Internet of things |
| **MDM** | Master Data Management |
|  |  |
| **PaaS** | Platform as a service is a cloud computing model that provides customers a complete platform—hardware, software, and infrastructure—to develop, deploy, manage, and run applications created by them or acquired from a third-party |
| **Pay-as-you-go** | Users can order cloud resources from a larger pool of available resources and pay for them on a per-use basis |
| **Private Cloud** | Cloud infrastructure provisioned for exclusive use by a single organization comprising multiple consumers, such as the business units within the organization |
| **Public cloud** | Users get access to servers, storage, network, security, and applications as services delivered by cloud service providers over the internet |
| **SaaS** | Software as a service is a cloud offering that provides users with access to a service provider’s cloud-based software |
| **SIP** | SaaS integration platforms |
| **TCO** | Total cost for ownership |
| **VM** | Virtual machine |
| **VPC** | Virtual Private Cloud |

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **ACL** | Access Control Lists |
| **AZ** | Availability Zones are distinct Data Centers with their own power, cooling and networking resources. These Zones can have names like DAL-09 or us-east-1. |
| **Bare-metal hypervisor** | Installed directly on top of the physical server. They're more secure, have lower latency, and are usually the ones you see in the market the most |
| **Block storage** | Is presented to compute nodes using high-speed fiber connections, which means that read and write speeds are typically much faster and reliable than with file storage |
| **CDNs** | Content Delivery Networks is a distributed server network that delivers temporarily stored, or cached, copies of website content to users based on the user’s  geographic location |
| **Cloud Region** | A geographic area or location where a Cloud provider’s infrastructure is clustered, and may have names like NA South or US East |
| **Containers** | Are an executable unit of software in which application code is packaged, along with its libraries and dependencies, in common ways so that it can be run anywhere, whether it be on desktop, traditional IT, or the cloud |
| **Data center** | A huge room or a warehouse containing cloud infrastructure |
| **Dedicated hosts** | Offer single-tenant isolation |
| **Direct Attached storage** | Or Local storage is storage which is presented directly to a cloud-based server and is effectively either within the host server chassis or within the same rack |
| **File storage** | Is typically presented to compute nodes as ‘NFS Storage’. NFS stands for Network File System and means that the storage is connected to compute nodes over a standard ethernet network |
| **Hosted hypervisor** | There's a layer of host OS between the physical server and the hypervisor. These hypervisors are less frequently used and mostly used for “end-user” virtualization |
| **HPC** | High-performance computing |
| **Hypervisor** | A piece of software that runs above the physical server or host |
| **IOPS** | Input/Output Operations Per Second and refers to the speed at which the disks can write and read data |
| **NFS** | Network File Storage |
| **Object storage** | Storage not attached to a compute node, rather it is accessed via an API |
| **Reserved virtual server** | Instances allow you to reserve capacity and guarantee resources for future deployments |
| **SDN** | Software Defined Networking |
| **Shared or Public Cloud VMs** | Are provider-managed, multi-tenant deployments that can be provisioned on-demand with predefined sizes |
| **Transient or Spot VMs** | Take advantage of unused capacity in a cloud data center |
| **Virtualization** | Process of creating a software-based or virtual version of something whether that be compute, storage, networking, servers, or applications |
| **VLANs** | Virtual Local Area Networks |
| **VM** | Virtual machines are software-based computers, based on virtualization technologies |
| **VPC** | Virtual Private Cloud |
| **VPN** | Virtual Private Networks |

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **API** | Application Programming Interface |
| **Application modernization** | Helps organizations accelerate their digital transformation, take advantage of new technologies and services, and become more responsive to changing market dynamics |
| **Cloud native application** | An application developed from the outset to work only in the cloud environment, or an existing app that has been refactored and reconfigured with cloud native principles |
| **Continuous delivery** | Delivering small, well-designed, high-quality increments of software to customers |
| **Continuous deployment** | Progressing each new packaged build through the deployment lifecycle as rapidly as possible |
| **Continuous integration** | Creating packaged builds of the code changes released as immutable images |
| **Continuous monitoring** | Monitoring with tools that help developers understand the performance and availability of their applications, even before they’re deployed to production |
| **Delivery pipeline** | An automated sequence of steps that involves the stages of Ideation, Coding, Building, Deploying, Managing, and Continuous Improvement |
| **DevOps** | Collaborative approach where business owners and the development, operations, and quality assurance teams collaborate to continuously deliver software |
| **Hybrid multicloud** | An open standards-based stack that can be deployed on  any public cloud infrastructure |
| **Microservices** | Break down large applications into their core functions |
| **Microservices architecture** | Approach in which a single application is composed of many loosely coupled and independently deployable, smaller components or services |
| **Monolithic architecture** | Approach in which a single application is built out of one piece of software |
| **Serverless** | Approach to computing that offloads responsibility for common infrastructure management tasks |
| **Service discovery** | Creates a roadmap for microservices to communicate |

**M5: Cloud Security and Monitoring, Case Studies, and Jobs**

Glossary

|  |  |
| --- | --- |
| **Term** | **Definition** |
| **Access group** | A group of users and service IDs is created so that the same access can be assigned to all entities within the group with one or more access policies |
| **Administrative users** | Create, update, and delete application and service instances, and need insight into their team members’ activities |
| **API keys** | Unique identifiers are passed into an API to identify calling application or user |
| **Application Performance Monitoring** **(APM)** | Measures application availability and performance, providing tools needed to troubleshoot issues in an application's environment |
| **Application users** | Users of the cloud-hosted applications |
| **AppSec** | Application Security |
| **Audit and compliance** | A critical service within identity and access framework used to validate implemented controls against policies |
| **Authentication** | Also known as identity service, it enables applications deployed to the cloud to authenticate users at an application level |
| **BYOK** | Bring Your Own Keys |
| **Client-side encryption** | Occurs before data is sent to cloud storage |
| **Cloud directory services** | Used to securely manage user profiles and associated credentials inside a cloud environment |
| **Cloud encryption** | Also known as the last line of defense, it encrypts data and provides robust data access control, key management, and certificate management |
| **Cloud monitoring solutions** | Assess data, application, and infrastructure behaviors for performance, resource allocation, network availability, compliance, and security risks and threats |
| **Cloud security** | Policies, technological procedures, services, and solutions designed to secure the enterprise applications and data on the cloud against insider threats, data breaches, compliance issues, and organized security threats |
| **Database monitoring tools** | Help track processes, queries, and availability of services to ensure the accuracy and reliability of database management systems |
| **Decryption key** | Defines how the encrypted data will be transformed back to legible data |
| **Developer users** | Authorized to read sensitive information and to create, update, and delete applications |
| **Encryption** | Scrambling data to make it illegible |
| **Encryption algorithm** | Defines the rules by which data will be transformed so that it becomes illegible |
| **Encryption at rest** | Protecting data while it is stored |
| **Encryption in transit** | Protecting data while it is transmitted from one location to another |
| **Encryption in use** | Protecting data when it is in use in memory |
| **Identity and access management** | Also known as access control, it helps authenticate and authorize users and provides user-specific access to cloud resources, services, and applications |
| **Infrastructure monitoring tools** | Identify minor and large-scale hardware failures and security gaps so that developers and administrators can take corrective action before problems affect user experience |
| **Key management services** | Help perform life cycle management for encryption keys that are used in cloud services or customer-build apps |
| **KYOK** | Keep Your Own Keys |
| **Multifactor authentication** | Adds an additional layer or authentication for application users |
| **Reporting** | Provides a user-centric view of access to resources |
| **Server-side encryption** | Occurs after cloud storage receives your data but before the data is written to disk and stored |
| **SSL** | Secure Sockets Layer |
| **TLS** | Transport Layer Security |
| **User and service access management capability** | Enables cloud application and service owners to provision and de-provision user profiles with minimal human interaction |

**Module 1 Glossary: Introduction to Application Development**

| **Term** | **Definition** |
| --- | --- |
| **AngularJS** | An open-source JavaScript framework for dynamic web applications |
| **Application Programming Interface (API)** | Code that allows two software programs to communicate with each other. |
| **Build Automation** | Allow you to download dependencies, compile code, package binary code, run tests, deploy to production. |
| **Build Automation Servers** | Execute build-automation utilities on a scheduled or triggered basis. |
| **Build Automation Utilities** | Generate executables by compiling and linking code. |
| **Continuous Integration/**\*\* Continuous Deployment (CI/CD)\*\* | A method for releasing code and integrating it into code that has already been developed in order to prevent the application from breaking throughout the app’s lifecycle. |
| **Django** | A framework for Python web development. |
| **Dynamic Content** | Data that is created each time a request is sent to a server. |
| **Endpoint** | The point at which an API connects with the software program. |
| **Frameworks** | Provide a standard way to build an application. Frameworks dictate architecture and program flow. |
| **IDE** | “Integrated Development Environment” Helps create and manage code. |
| **Inversion of Control** | A predefined workflow where the developer is not in full control of how the application operates. |
| **JavaScript Framework** | An application framework written in JavaScript to create responsive sites. |
| **LESS** | “Learner Style Sheets” add more style and functions to CSS. |
| **less.js** | A JavaScript tool that converts LESS styles to CSS. |
| **Libraries** | Reusable collections of code |
| **Opinionated** | Frameworks that have a lot of control are sometimes considered “opinionated”. |
| **Package Managers** | Coordinate with file archivers to extract packages. Verify check sums and digital certificates. Locate, download, and install updates of existing software from a repository as well as manage dependencies. Common package managers include the following: Debian Package Management System (DPMS), Red Hat Package Manager for Linux, Chocolatery for Windows, Homebrew and MacPorts for MacOS. |
| **Packages** | Archive files that include app files, instructions for installation, and metadata. |
| **React.js** | A JavaScript framework developed by Facebook that helps build and drop elements onto a page. |
| **Responsive Design** | Design technique that automatically resizes a display to adapt to a specific screen size. |
| **Route** | Allows front-end client to plug into correct socket on the backend. They are the paths that network traffic takes from a virtual machine (VM) instance to other destinations. |
| **SASS** | “Syntactically Awesome Stylesheets” are an extension of CSS. |
| **Static Content** | A display of data that has been previously stored on a server. |
| **Version Control** | Allows you to revert to earlier versions of code, resolves conflicts between the same files, and split and merge different code branches. |
| **Vue.js** | A community-based JavaScript framework focused on UI. Includes UI components such as buttons and other visual elements, and is both a library and a framework. |

| **Term** | **Definition** |
| --- | --- |
| **DOM Tree** | “Document Object Model” is the data representation of the objects that comprise the structure and content of a document on the web. |
| **XML** | An “eXtensible Markup Language” Designed to store and transport data allowing users to define their own markup languages, primarily to displaydocuments on the web. |
| **XHTML** | An “eXtensible Hypertext Markup Language” similar to HTML but with stricter formatting rules. |
| **Web Storage APIs** | APIs that allow data storage in a browser. |

| **Term** | **Definition** |
| --- | --- |
| **Component Framework** | Component frameworks provide pre-styled components and templates which are easy to add to any website. |
| **CSS** | "Cascading Style Sheet"s is a style sheet language that describes how HTML elements are displayed​. It is the design that is layered over the top of an HTML web page​. |
| **Fixed Layout** | A fixed layout is a layout where ​you specify the height and width of elements, and those values remain the same regardless of which operating system or browser you use to access the website. |
| **Fluid Layout** | A fluid layout is a layout in which ​the height and width of elements is flexible ​and can expand or contract based on the browser window, the operating system, and other user preferences. |
| **Utility Framework** | The utility framework provides utility classes that are scoped to individual CSS properties, which helps in building custom designs in HTML files. |