

Yvonne V. Richardson  
P. O. Box 13430  
Mill Creek, WA 98082  
yvonne.richardson@comcast.net

about 2350 words

## Cloud Computing

By Yvonne V. Richardson

### Cloud Computing

The phrase "cloud computing" originates from the flow chart cloud symbol to symbolize the Internet. Cloud computing is a model for enabling access to networks, computers, storage, applications and services. It is partially based on the idea of LPARs, or logically partitioning a physical computer into a series of virtual smaller ones. A logical partition, or LPAR, is a subset of the processor hardware that contains resources (processors, memory, and input/output devices) and can operate as an independent system. Multiple logical partitions can exist within a hardware system.

Cloud technology takes this idea one step farther in that the technologist creates partitions, or virtual machines, to complete a task. Technologists rent space for this from companies that allow access to their physical data centers. Those companies also subscribe to this virtual philosophy, so that the space that they make accessible to cloud technologists is not constrained to a specific data center. Consequently, renting space in the cloud is like renting a locker in a building, except when you are done with the virtual computer, you can delete it, whereas a locker does not cease to exist.

Another difference between physical data centers and cloud computing is that cloud technology is defined as CSN - computer, storage, and networking, which splits the definition of LPAR in half. One half combines the LPAR processor and memory under the heading of "computer" - CPUs and RAM. The other half of the LPAR definition is also split in half. Input/output storage devices might be attached to the virtual computer in the same way that we used to install internal disk drives. If storage was external, you may have used a cable instead of a network. Remember the NIC card that you could assign a port address in addition to an IP address? Nowadays, in the cloud, storage could have a networking address that is different from the virtual computer that uses it.

A long time ago I was a JCL jockey in an IBM shop, so I understand the perspective that cloud technology tries to convey. A cloud technologist provides an environment of computers, storage, networking, and operating systems in which application developers can work. The developers, in turn, create tools and processes for their clients. The cloud technologist's purview stops where the developer's role begins. The cloud technologist is on the outside of the cloud, while the application developers and their clients are on the inside. In other words, a cloud technologist might create a site on which you can blog to your heart's content, but the technologist does not interact with the blog's content.

## Acquiring Cloud Skills

I recently completed a certificate in a cloud technology bootcamp. A bootcamp is a short, intensive, and rigorous course of training, in a format that differs from a college quarter. This bootcamp was a little short on the rigor, as traditionally defined by educational institutions, but it makes up for that by being online in a 24-7 format. The program's goal was to quickly develop the necessary cloud skills for individuals who want to increase their employability in an ever-changing job market.

The class relied heavily on open-source software, which is code that is designed to be publicly accessible--anyone can see, modify, and distribute the code as they see fit. Tools were available for free or for a small fee, which is reasonable for an educational experience. The class really values learning by building and doing, and with more practice, the average individual can converse about and use cloud tools. If you have a computer that uses any major operating system and has access to the Internet, you can configure and monitor cloud services.

## Speed-Reading the Cloud

As is customary for a bootcamp or any other immersion experience, there was way too much information presented for the timeframe, and it felt like I had registered for five or six college courses and then skimmed them until it was time for midterms. At that point, I was supposed to consider the topic to be complete and continue to the next one. The pace and the depth provided some bewilderment, but the course taught the basics of many tools that are vital to succeeding in the cloud computing world.

Like speed reading, which changes how you select information to process, this course presented concepts which illustrate the main ideas associated with each real-world software tool. The conceptual overview included enough tools for a cloud technologist to provision businesses and serve customers. Individuals who already have a cloud computing background, or data center/machine room experience, would appreciate the overview, which included basic installation and validation of cloud components.

Another reason for the speed-reading approach was to teach you that information that you do not have, but that you require to complete a task, is probably available in online manuals. By the time you have questions, you know how to research answers in a reasonable amount of time. Concepts like CI/CD and database partitioning were reserved for more advanced courses.

The course was vendor-neutral, so as to provide exposure to tools and solutions from many companies. It was aware of, but did not exactly try to avoid, the turbulence that happens when pioneering software companies carve out niches and then compete. For example, Chef is a configuration management tool that streamlines the task of configuring and maintaining a company's servers. It was the full-meal deal for a long time, until Terraform ate a large piece of the Chef market.

As another example, Docker is an open-source platform for building, deploying, and managing containerized applications. It became one of the leading full-service tools, and then Kubernetes came along and took a bite out of Docker. Kubernetes is used to coordinate and schedule containers, allowing for application upgrades without interruption of service. Chef, Terraform, Docker, and Kubernetes are supported by a number of vendors, but the Docker-Kubernetes-Terraform toolset seems to be the one that provides the most automated cloud services to businesses.

## Cloud Control

In the cloud, the command-line interface is one of the primary tools that is used to provision virtual machines and containers by running tasks, or by combining tasks into scripts. The interface uses languages like bash or PowerShell, and it enables an individual to understand and control what they are doing in the cloud. The results are similar to the automated products and services that cloud vendors offer on their Web pages. Any scripts that used tools from one vendor can be customized to use tools from another vendor.

It is good practice to learn one tool reasonably well, and then explore similar ones after you have mastered the basic concepts. For example, the installations of Octopress and WordPress were reasonably straightforward, but the installation of other blogging tools is left as an example for the user.

## Cloud Tool Map

This outline categorizes over 80 basic cloud components. It includes methodologies and perspectives for specific tools, or you can scan the sections for the information you need to build your cloud. This is by no means intended to be a comprehensive list, but it may help you make sense of the bewildering array of technologies that are displayed in Internet searches and on cloud provider Web sites.

### 1. Services and Service Models

*These service models illustrate how cloud components are related, and may assist new technologists to select tools and build network models with which to support customers.*

- **Infrastructure as a Service (IaaS)**  
Infrastructure as a Service (IaaS) is first. It is on-demand access to cloud-hosted servers, storage and networking, and is required for running applications and batch processing in the cloud.
- **Platform as a Service (PaaS)**  
Platform as a Service (PaaS) is second. PaaS is the underlying infrastructure for application or service development. PaaS is often built on top of an IaaS platform, and may include items like databases so that the applications can run. Platforms like Heroku or Apache can handle operations for DevOps so that the devs can focus more on development and rapid deployment.
- **Software as a Service (SaaS)**  
Software as a Service (SaaS) is built on top of PaaS. Cloud-hosted application software includes items like email, Office 365, or Salesforce.

### 2. Security

*How much security is appropriate for your network is implementation-dependent.*

Certification (HTTP vs HTTPS)	Keys and key pairs
Credentialing	Secure Shell (ssh)
Domain Name System (DNS) sellers	Secure Sockets Layer (SSL)
Domains	Transport Layer Security (encryption)
Internet Corporation for Assigned Names and Numbers (ICANN)	Virtual Private Networks

### 3. Basic Cloud Services

*These services can be provided without a large or complex cloud network.*

Blogging	Email Hosting
Cloud storage: Amazon S3, Digital Ocean Spaces, Google Cloud, Microsoft Azure Blob	Static Web site Generators: Hugo, Octopress, Pelican
Content management service	Virtual machines
e-Commerce	Containers (similar to virtual machines)

### 4. Network Infrastructure

*When using more than one computer, these components can be used in any network, whether LAN, WAN, or cloud. This includes replication across content delivery networks.*

Content Delivery Networks	Ports
Domain Routing (CIDR)	Proxies
Gateways	Routing Tables
Load Balancing	Subnets
Network Address Translation	Virtual Private Clouds
Network security groups	Virtual Private Network (VPN)

### 5. Source Sharing

*Source code repositories allow developers to share and update code over the Net.*

- GitHub
- DockerHub for images (similar to ISOs)
- BitBucket

### 6. Operating Systems

*According to Fossbytes.com, in 2015, Ubuntu Linux was the most popular cloud operating system. The ones we used most often during the bootcamp were Ubuntu or Ubuntu-based, although operating systems like Google Chrome and Windows Azure are also used in the cloud.*

### 7. Programming Languages

*Some tools expect specific languages to be installed on the computer on which the tools will run. You may or may not need to learn the individual languages. A quick Internet search will display a definition of the tool from Wikipedia or from the manufacturer.*

Bash	PHP (aka Perl)
Best Practices	PowerShell
Bundle (Bundler)	Python
Domain-Specific Language (DSL)	Rails
Editors: VS Code, gedit, nano, Notepad, vi	Ruby
Helm	RubyGems
Jekyll	Spring
JSON	YAML

## 8. Databases

*Some tools expect specific databases to be available. You need to locate the ones that are most compatible with the front end that your AppDevs will use: sql-based, key-value pair-based, and others. Only the databases that were required for simple cloud applications were used. Advanced configuration, partitioning and tuning were outside of the scope of the bootcamp course.*

- MySQL
- Postgres

## 9. Applications and Software

*Software as a Service is a category of tools that is primarily on top of the platforms that you have built with components in items 4 through item 8, although in some instances, the tools might be considered one of the previous categories.*

- Content management systems – WordPress, et al
- File Conversion – image- and video-based
- Media Streaming
- Site reliability and availability tools
- Web Servers – Apache, NGINX

## 10. Maintenance

*Data centers must be planned and maintained. This includes hardware and virtual containers as well as applications, data, and persistent storage.*

Architectural maps or diagrams	Health checks
Backups	Logging
Batch processing	Methodology
Capacity planning	Monitoring
Cheat sheets and manuals	Run books and playbooks
Configuration scripts	Scaling
Documentation	Snapshots and Restores

## 11. Configuration Management and Automation

*Defining cloud infrastructure as code allows you to programmatically split applications into roles, and then assign the roles to separate nodes within the network. These tools are used to plan, document, and execute the scripts that define the network, services, applications, connectivity, and replication.*

- Ansible
- Chef
- Docker
- Terraform
- Docker Compose

## 12. Platform Providers

*This is a list of businesses that provide automated tools and services from the other items in the tool map. Their services can be combined with your blueprints and scripts. You can also replace your scripts by purchasing services and support from these companies.*

Amazon Web Services (AWS)	Microsoft Azure
Amazon Elastic Cloud (EC2)	Digital Ocean
Google Cloud	

## Navigating the Tool Map

The list of cloud tool categories becomes a map that you can navigate to build network, and build cloud technologist skills, by first understanding item 1. The categories in the map depend on each other, so you may find yourself reading ahead, or using objects that will be described in a later category.

You can create one virtual machine that can be shared on the Internet by performing items 2 and 3 together. When you want to build a network of more than one computer, use these concepts to perform item 4. This may be familiar to those who have LAN experience. When you are comfortable building small networks, then go on to items 5 and 6 by selecting operating systems from shared sources for your cloud computers. This is a good point at which to experiment with what has been made available to you for in-depth learning.

At that point you will have developed a comfort level with IaaS, and are ready to go on to PaaS with items 7 and 8. These platforms will contain rudimentary services that you make available from your Internet address. For example, a WordPress platform requires mysql, which you are aware of as a cloud technologist, but the database is not exposed to users of your WordPress site.

Item 9 is SaaS, and is the point where you have built an entire cloud network for serving customers whose needs are more complex than the business requirements that can be met using only items 1 through 3. You will also have had the opportunity to make a list of auxiliary items in your DevOp environment, which should resemble item 10. You will have processes that you repeat, or scripts that you can re-run, to maintain the cloud sites you have built, or to create larger cloud environments. Item 11 is a close-up of the tools that you are using to automate site building and deployment.

Finally, item 12 lists cloud vendors that provide any or all of the elements from previous items on the list. Based on your experience, you can select which tools you want to build and which services you want to purchase when you plan and create the virtual data center that is your presence in the cloud.