**Infrastructure as a Service (IaaS) Vs. Infrastructure as Code (IaC)**

Zakeera Hasunudeen, Samantha Huyck-Trout, Deepa Natarajan, Steve Studsdahl

Group Assignment, Group 5

CS 515: IT (Information Technology) Infrastructure

Bellevue University

James Krohn

June 03, 2021

**Infrastructure as a Service (IaaS) Vs. Infrastructure as Code (IaC)**

Infrastructure as a Service (IaaS) and Infrastructure as Code (IaC) are complementary technologies that allows for rapid deployment of systems with repeatable results. When looking at IaaS, cloud computing immediately comes to mind. IaaS is nothing more than gaining access to computing systems in which you pay for the consumption of the resources that you use. These resources typically include compute, network, and storage access. IaC is a mechanism that allows configurations to be deployed to hardware regardless of whether it is on-premises hardware or obtained through the IaaS model. IaC allows for the automation of updating and changing system configurations requiring less time and effort to provision and maintain systems.

**Infrastructure as a Service**

IaaS is typically an on-demand, self-service system that provides resources in the form of compute, network, storage, software, and other resources when needed by a consumer. These resources are made available using automation tools that provision resources according to parameters defined by the consumer that fall within the capabilities of the host of the IaaS. These deployed resources are generally broadly accessible and can be secured as appropriate to meet the needs of the customer who is consuming the resources.

IaaS generally provides a pool of resources that the hosting provider makes available to customers to provision to meet the customer’s needs. The ability for a customer to provision resources as they need them provides the rapid elasticity that businesses often need to meet the needs of the moment to provide services to their customers or even internal users. Provisioned resources can be monitored, controlled, and reported on which can in turn be used to bill for resource usage.

Infrastructure as a Service can lower the cost of resources for an organization as the cost is incurred for the resource usage at the time of usage rather than having to pay for the hardware up front. IaaS also shifts the budget from a capital expenditure (CAPEX) model to an operational expenditure (OPEX) model. The nature of IaaS hardware allows for customers to have access to newer, higher performance equipment without having to pay the premium price to own the resource outright.

The on-demand nature of IaaS allows for greater scalability to grow system capacity as the need arises and allows the same capacity to be scaled back during times when there is less demand required of the systems. The on-demand nature of IaaS also allows customers to respond quicker to the capacity needs of the business so that they can in turn provide services that meet the expectations of their customers.

IaaS is also a great fit business continuity in the event of a disaster. If backups are stored with an IaaS provider, a scenario requiring restoration of services from on-prem systems can be done quickly by provisioning systems within the IaaS environment and restoring the data and application backups to these newly provisioned systems.

While IaaS provides many benefits to consumers of this model, there are also some drawbacks that must be factored into the decision to use IaaS. The first of which is access to the IaaS systems. These systems are normally accessed across the Internet which make the connectivity subject to any issue in the path across the Internet when reaching the systems. Should Quality of Service be required, the Internet connection will not provide that which then requires establishing a direct connection into the IaaS provider. (IBM Cloud Learn Hub).

Additional difficulty can be experienced if ever the need arises to migrate to a different IaaS provider. Not only must applications be provisioned but all data must be transmitted to the alternate provider which will require some sort of connectivity between the two IaaS providers. Depending on the amount of data that needs to be transmitted, this can be a lengthy and expensive process.

**Infrastructure as Code**

Infrastructure as Code is the ability to provision and configure systems in an automated fashion to return consistent, known results each time. IaC is especially important when it comes to cloud computing or IaaS. It is also beneficial with on-premises systems to know that new systems are provisioned correctly and according to company standards and best practices. IaC makes it possible to manage resources more efficiently and proactively and reduces the number of errors that can be introduced through typos and other human error.

IaC brings the concept of version control to system provisioning. The code (scripts) that are used with different automation tools can be managed within a version control system such as Git. Using a version control system allows for the scripts to be continuously updated to reflect any changes to policies or best practices over time and allows for changes to the scripts to be audited and backed out if necessary.

DevOps benefits from IaC as it enables validation and testing of infrastructure code throughout the DevOps pipeline and provides confidence in the progression of system development and with what gets published into a production environment. Coupled with version control, issues that are experienced can be backed out rather quickly as the changes that were made are documented in version control system.

Infrastructure blueprints are easily created with the use of IaC. Specific policies that need to be incorporated into systems being deployed are easily defined within the scripts and ensure that best practices are adhered to without overlooking a portion of the configuration of the system.

There are three categories of tasks that IaC can be broken down to and are infrastructure provisioning, infrastructure configuration, and application deployment. These tasks are applied in two different phases of the life cycle of the deployment, the initial setup phase and the maintenance phase.

Infrastructure provisioning is tasked with spinning up new servers, performing network configurations including deployment and configuration of any network resources such as load balancers and firewalls. All initial infrastructure provisioning tasks are handled through this task.

The infrastructure configuration task is responsible for system specific configurations to support the applications being deployed. If web servers are being deployed, part of the configuration would include load balancers to spread requests across multiple servers if the need was there and mange any firewall configurations to allow connections to be established to the web server on the specific ports that the application calls for. Application deployment, as the final task, is responsible for the installation of any software and required software components to bring the system into a production ready state. This could be done through installation of all software components on a system directly or through the deployment of containerized applications running within a Docker environment.

These three tasks are run initially during the setup phase of a system or environment. Once the system has passed through this phase, ongoing maintenance phases will repeat the processes to keep systems up to date with current software versions appropriate to what was deployed and handle any configuration changes that may be required due to either changes in policy, best practices, or functionality. IaC requires different tools for different functions. Initial deployment of systems can be handled by a program called Terraform, but this tool is not capable of handling the ongoing maintenance needs of the systems. To handle those tasks, tools such as Chef, Puppet, or Ansible can be used.

IaC brings to the table reproducible results. This is critical to achieve efficiencies when deploying and maintain systems and provides a mechanism to scale support to larger number of systems with fewer resources. The scripts used in the IaC processes provide the consistency required to standardize configurations and ensure that there is a known good base to start from.

When using IaC to provision systems, tests and validations are required to ensure that all components are deployed properly and are functioning. Tests that fail should have a way of alerting to the fact that a failure has occurred. (Morris, K., 2016)

**Table 1. Pros: Infrastructure as Code Vs Infrastructure as a Service**

|  |  |
| --- | --- |
| **Infrastructure as Code** | **Infrastructure as Service** |
| Reproducible Infrastructure | Lowest Infrastructure cost |
| Accountability | On demand scalability and high performance |
| Consistency | Secure Infrastructure and higher resource utilization |
| Repeatability and Idempotency. | Respond quicker to shifting business conditions. |
| Enable design, deploy declarative and imperative DevOps paradigms. | Increase stability, flexibility, reliability, and supportability. |
| Speed, Simplicity and Security | Capable to match consumption to demand. |
| Provides flexibility to build mutable and immutable ecosystems. | Higher the utilization, higher the return on assets |
| Automated tasks and Code language flexibility | Improves business continuity and disaster recovery. |
| Scalability | Focus on core business and deliver new apps faster. |

**Table 2. Cons: Infrastructure as Code Vs Infrastructure as a Service**

|  |  |
| --- | --- |
| **Infrastructure as Code** | **Infrastructure as Service** |
| Complexity | Portability of applications is a challenge. |
| Automated Failures | Does not work without internet service. |
| Precarious Security | Limited user privacy, Customization and Virtualization |
| Additional complexity with new type of code additions. | Extension of internal security models is difficult causing potential security flaws. |
| Working collaboratively | Maturity of system management tools |
| Hard to maintain all the code. | Few Technical problems cannot be directly fixed by business. |
| Longer turn-around time for every change | Keeping tab of regular obligated upgrade and maintenance |
| Extra cost due to reproducible and repetitive code | Unable to clone. |
| Integrating it with the CI/CD pipeline | Performance Issues |

**Application**

Application of IaaS depends on the nature of the workload. It may vary based on the factors like computing capacity, back-up services, security needs, network bandwidth and other QoS requirements. The application can be broadly divided into three groups-server centric, client centric and mobile centric.

Server centric workloads include websites for social networking, scientific computing for atmospheric modeling, Enterprise software like SAP, online financial services like banking or insurance, E-commerce. For these services the key Quality of metrics focuses more on capacity, security, high network bandwidth. When it comes to client centric application it includes productivity applications like word processing, Development and testing environments where network bandwidth, data back up and security becomes the key metrics. Mobile centric includes the rich mobile applications with high availability as priority.

IaC can be applied to automate the infrastructure either as mutable or immutable infrastructure. While applying IaC solution, the automation can be approached as declarative or as imperative. In the first approach the final state of the infrastructure is declared at the start whereas, in the later approach the automation scripts provide the solution one step at a time.

Mutable Infrastructure can be modified or updated after it is provisioned. This gives the flexibility for the developers to modify the environment, on the flipside it compromises the important characteristics of Infrastructure version control tracking system. To overcome this drawback, IaC is commonly applied as Immutable infrastructure. This approach discourages the modification in the code, instead it encourages complete replacement with new infrastructure. It eliminates the configuration drift and maintains consistence between the test and deployment environment. (Amies *et.al.,* 2012).

**Use Cases**

Using IaaS allows additional servers to be deployed rapidly to address the traffic demands of web servers. This is especially critical for online retailers during shopping seasons such as Christmas. The additional servers fall in line with the existing functionality of the existing web servers to ensure that the web site(s) respond quickly to customers and are available to customers.

IaC helps to ensure that accurate configurations of software and all required components are in place for the web site(s) to function. Cost reductions are achieved by not having to purchase the physical hardware and allocate space for the hardware onsite. As demand is reduced, servers can be turned down to reduce the operational costs of the web servers. As servers are turned down, the cost of operating the additional hardware is eliminated which lowers the cost of operation. The IaaS model also helps to provide an efficient use of resources as there are less idle resources to account for both in operational expense and depreciation.

IaaS is a great way to develop systems/software without the expense of outright purchasing hardware to run the software on. This can allow commercial software to be evaluated with lower expenses than if hardware had to be purchased. Development environments can be setup allowing for continued enhancements of systems without impacting production systems. Once new systems are ready to roll out, IaaS and IaC can permit a rollout of the system in a repeatable manner to expand resources to meet the capacity demands of the production system.

**Table 3. Best Practices**

|  |  |
| --- | --- |
| **Infrastructure as Code** | **Infrastructure as Service** |
| Make code your single source of truth – Codify everything. | Model business-critical IT services |
| Use little documentation or none as possible. | Complete and up-to-date IT monitoring |
| Continuously test, integrate, and deploy. | Align IT and business teams with meaningful dashboards and reports and proactively share status with your end users. |
| Maintain version control. | Secure privileged access |
| Make your infrastructure code modular. | Monitor, Trend and Alert on Cloud Resource Consumption |
| Make your infrastructure immutable when possible. | Integrate Metrics & Flows Data for a Complete View |

**Summary**

IaC is a crucial part of the Agile Development and DevOps enabling Developers to get more involved in defining configuration and Ops teams to get involved earlier in the development process. IaC can deploy and manage all types of computer resources including servers, databases, services, virtual networks, permissions, block devices, and cloud provider offering.

IaaS is a standardized, highly automated offering which provides businesses with a platform on which software can be developed and deployed with scalable resources and metered by use based on the number of resources allocated and consumed.

IaaS is like Software-as-a-Service (SaaS) and Platform-as-a-Service (PaaS), in that it provides services to consumers over the Internet. However, out of the three offerings, IaaS takes care of the essentials, allowing users to manage the rest.

**Reference**

IBM Cloud Education. (n.d.). *Infrastructure-as-code*. Retrieved April 30, 2021, from <https://www.ibm.com/cloud/learn/infrastructure-as-code>

Ravikanth.C.,(2016, January 15). *DevOps, Infrastructure as Code, and PowerShell DSC: The Introduction.* PowerShell Magazine. <https://www.powershellmagazine.com/2016/01/05/devops-infrastructure-as-code-and-powershell-dsc-the-introduction/>

Peter Mell and Timothy Grance (September, 2011). [*The NIST Definition of Cloud Computing*](https://dx.doi.org/10.6028/NIST.SP.800-145) (Technical report). National Institute of Standards and Technology: U.S. Department of Commerce. [doi](https://en.wikipedia.org/wiki/Doi_(identifier)):[10.6028/NIST.SP.800-145](https://doi.org/10.6028%2FNIST.SP.800-145). Special publication 800-145.

Alex Amies, Harm Sluiman, Qiang Guo Tong and Guo Ning Liu (2012*).* [*Developing and Hosting Applications on the Cloud: Develop Hosting Applica Cloud*](https://books.google.com/books?id=4gwIYbtTH5MC)*.* Pearson Education. [ISBN](https://en.wikipedia.org/wiki/ISBN_(identifier)) [978-0-13-306685-2](https://en.wikipedia.org/wiki/Special:BookSources/978-0-13-306685-2).

TechWorldWithNana.(2020, August 28). *What is Infrastructure as Code? Difference of Infrastructure as Code Tools*. YouTube: <https://www.youtube.com/watch?v=POPP2WTJ8es>

Kief Morris. (2016). *Infrastructure As Code: Managing Servers in the Cloud*: Vol. First edition. O’Reilly Media. <https://search-ebscohost-com.ezproxy.bellevue.edu/login.aspx?direct=true&db=edsebk&AN=1249842&site=eds-live>.

IBM Cloud Learn Hub. (n.d.). *IaaS (Infrastructure-as-a-Service)*. Retrieved May 10, 2021, from <https://www.ibm.com/cloud/learn/iaas>

Agustin Romano (2020, October 22) *Infrastructure as Code vs. Infrastructure as Software* <https://caylent.com/iac-vs-ias>