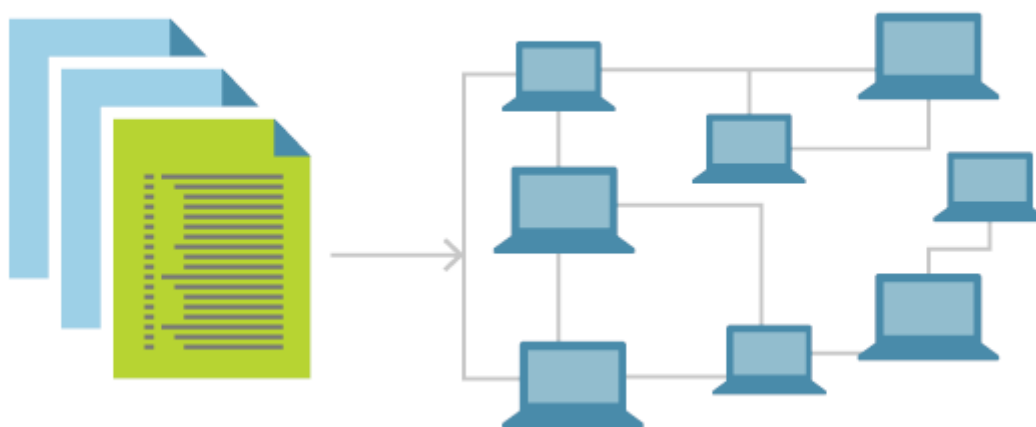


What is Infrastructure as Code?

04/04/2017 • 2 minutes to read •  

By: Sam Guckenheimer



Infrastructure as Code (IaC) is the management of infrastructure (networks, virtual machines, load balancers, and connection topology) in a descriptive model, using the same versioning as DevOps team uses for source code. Like the principle that the same source code generates the same binary, an IaC model generates the same environment every time it is applied. IaC is a key DevOps practice and is used in conjunction with [continuous delivery](#).

Infrastructure as Code evolved to solve the problem of *environment drift* in the release pipeline. Without IaC, teams must maintain the settings of individual deployment environments. Over time, each environment becomes a *snowflake*, that is, a unique configuration that cannot be reproduced automatically. Inconsistency among environments leads to issues during deployments. With snowflakes, administration and maintenance of infrastructure involves manual processes which were hard to track and contributed to errors.

Idempotence is a principle of Infrastructure as Code. Idempotence is the property that a deployment command always sets the target environment into the same configuration, regardless of the environment's starting state. Idempotency is achieved by either

automatically configuring an existing target or by discarding the existing target and recreating a fresh environment.

Accordingly, with IaC, teams make changes to the environment description and version the configuration model, which is typically in well-documented code formats such as JSON. The release pipeline executes the model to configure target environments. If the team needs to make changes, they edit the source, not the target.

Infrastructure as Code enables DevOps teams to test applications in production-like environments early in the development cycle. These teams expect to provision multiple test environments reliably and on demand. Infrastructure represented as code can also be validated and tested to prevent common deployment issues. At the same time, the cloud dynamically provisions and tears down environments based on IaC definitions.

Teams who implement IaC can deliver stable environments rapidly and at scale. Teams avoid manual configuration of environments and enforce consistency by representing the desired state of their environments via code. Infrastructure deployments with IaC are repeatable and prevent runtime issues caused by configuration drift or missing dependencies. DevOps teams can work together with a unified set of practices and tools to deliver applications and their supporting infrastructure rapidly, reliably, and at scale.

Learn more about Infrastructure as Code with [Azure Resource Manager](#).



Sam Guckenheimer works on Microsoft Azure DevOps team. He acts as the chief customer advocate, responsible for strategy of the next releases of these products, focusing on DevOps. He has written four books on DevOps and Agile Software practices.

Is this page helpful?

 Yes  No
