Enforce Secure Automated Deployment Practices through Infrastructure as Code

Executive summary

Infrastructure as code (IaC), baselines, and golden images are terms growing in usage across the cloud industry. These terms refer to templates that are used to deploy resources across on-premises and cloud infrastructures. IaC uses code to automate the deployment of compute, network, and storage services, as well as security policies (often denoted as policy as code). The terms baselines and golden images are often used interchangeably. They both use predefined templates to serve as starting points for secure system deployments.

Languages and formats used to define IaC templates vary by vendor, but are written to be human readable. Cloud service providers (CSPs) have built-in IaC services that can be used to deploy their specific resources. Open source and commercial IaC tools that are vendor-agnostic are also available for on-premises and cloud deployments.

This cybersecurity information sheet outlines key benefits of IaC and practices that should be considered before and after deploying IaC templates.

Figure 1: Benefits of using IaC

Why use IaC?

IaC is designed to address development and security issues that arise during the software development lifecycle, including, but not limited to, environment drift, lack of

reusability, and delayed discovery of security misconfigurations. The following sections outline how using IaC can alleviate common trouble areas. [1], [2]

The sections list several related MITRE ATT&CK® and MITRE D3FEND™ threat and defensive techniques. These are meant as representative examples, but are not intended to include all possible related techniques. [3], [4]

Elimination of manual deployments

Manually deploying cloud resources is not only time consuming, but prone to human errors that can lead to misconfigurations and security gaps. With IaC, resources are defined in a single location and included as part of the continuous integration/continuous delivery (CI/CD) pipeline. This saves time on repeatable tasks that are required to be deployed in multiple environments. IaC can also be combined with policy as code to ensure resources are vetted before deployment and force deployment failures if components are not properly configured. If used with a version control system, IaC also provides auditable logs of changes to template files.

Resource protection

Manual deployment also leads to difficulties tracking cloud resources. Not being able to properly track resources can lead to shadow IT, resulting in unmonitored assets, increased costs, and data breaches.

ATT&CK Tactic	Technique
Defense Evasion	Impair Defenses [T1562]
Defense Evasion	Modify Cloud Compute Infrastructure [T1578]

D3FEND Tactic	Countermeasure
Application Hardening	Application Configuration Hardening [D3-ACH]
Platform Monitoring	Endpoint Health Beacon [D3-EHB]

Immutable infrastructure

IaC can be leveraged for immutable infrastructures, which are built to exact specifications and require any modifications to be made through deployments of new IaC templates rather than on resources directly. Using immutable infrastructures reduces the risk of *ghost*, or unmonitored, assets, improving threat detection.

Drift detection

Many IaC tools perform drift detection to report when modifications have been made to infrastructure resources outside of what is defined in the template. Drift occurs when resources are updated manually. Drift detection can be run manually or set to run periodically. Drift detection alerts on any infrastructures altered after an IaC deployment. It provides information regarding the specific resource where changes were detected.

Tagging and resource standardization

Tagging is used to assign metadata to cloud resources and allows for easier grouping and monitoring of resources. These tags can be anything, including security classifications, resource owners, and environment descriptions. Using IaC to deploy resources can ensure the resources are defined in such a way that they are automatically tagged upon creation. This promotes resource standardization by ensuring grouped resources that require maintenance, such as patching, can be done in a unified fashion. Resource standardization ensures all resources are in compliance with organizational and regulatory policies.

Disaster recovery

IaC can assist in timely disaster recovery efforts. Infrastructure resources are already defined in template files. In the event infrastructure needs to be rebuilt, IaC templates can be redeployed versus rebuilding each resource manually if IaC template backups are properly stored and available.

Access control

Resources deployed in the cloud often require explicit allow actions to be accessible by users or other resources. The task of creating identity and access management (IAM) policies that define each action that can be taken on a resource can become cumbersome, especially when required for multiple accounts, and often results in more access being granted than necessary. IaC can provide higher assurance that the proper permissions are granted by defining reusable policies in a single location that provide just enough privilege for users of cloud services. Refer to the joint cybersecurity information sheet (CSI) Use Secure Cloud Identity and Access Management Practices for more information on implementing secure IAM policies.

ATT&CK Tactic	Technique
Resource Development	Compromise Accounts (Cloud Accounts) [T1586.003]
Persistence	Account Manipulation [T1098]
Discovery	Permission Groups Discovery: Cloud Groups [T1069.003]

D3FEND Tactic	Countermeasure
Operational Activity Mapping	Access Modeling [D3-AM]
Credential Hardening	User Account Permissions [D3-UAP]

Deployment considerations

The following sections provide actions that should be considered as organizations are creating IaC templates before and after deployment.

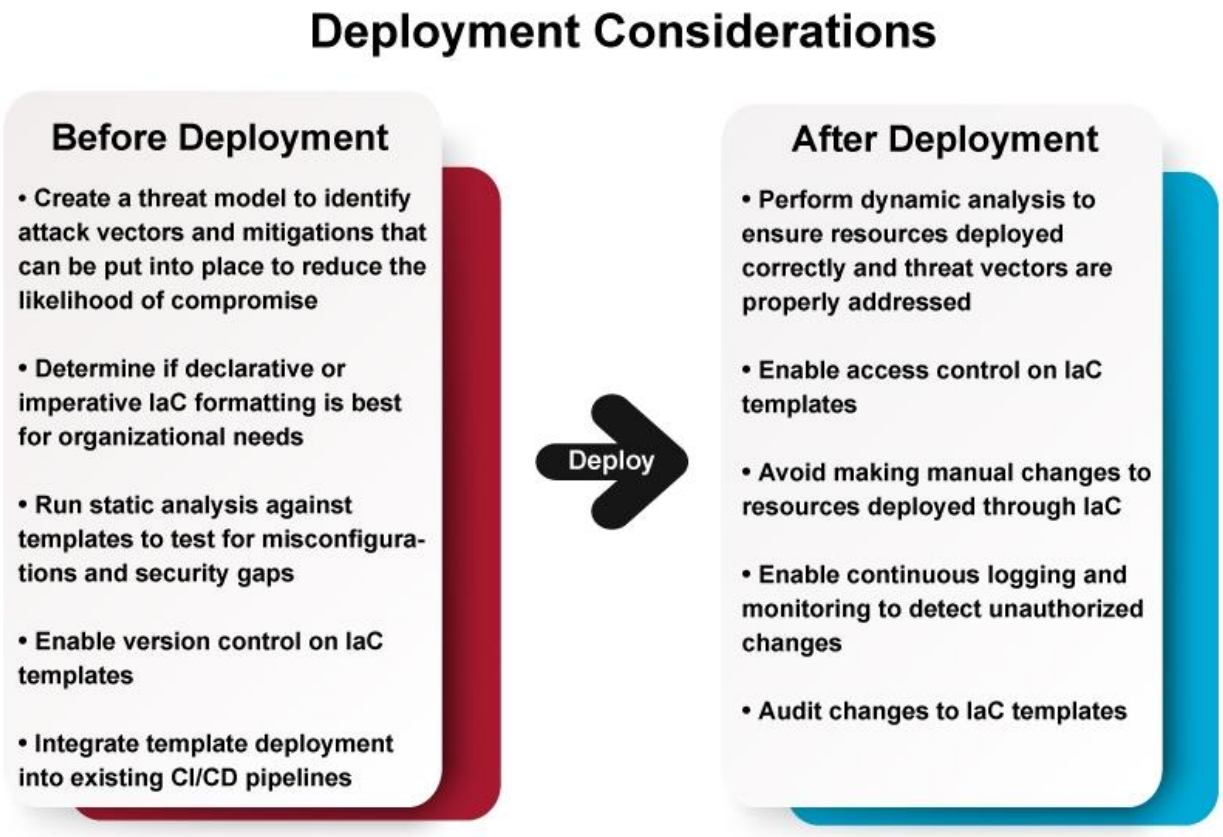

Figure 2: Deployment considerations

Before deployment

Create a threat model

Threat modeling is recommended before drafting IaC templates to understand threats to IaC templates, possible attack vectors for resources being created, and security controls that could prevent threats from coming to fruition, such as access controls and versioning. Threat modeling helps ensure a proactive approach to security rather than a reactive one by ensuring all gaps are covered before the resources are live. The MITRE ATT&CK framework can be used during this process to identify common adversary tactics and techniques that can be used against environments and mitigations that can be implemented to reduce the risk of adversary success. [3], [4]

D3FEND Tactic	Countermeasure
Operational Activity Mapping	Operational Risk Assessment [D3-ORA]

Determine the appropriate IaC style

IaC can be written in two formats: declarative and imperative. Determining which style works best depends on organizational needs and familiarity with resources being deployed.

Declarative IaC templates are written to define the desired end-state of resources being provisioned and is used in many of the popular IaC tools available today. It does not require extensive programming knowledge, is reusable, and gives the IaC tool control of how the resources are deployed.

Imperative IaC templates are written to describe how resources should be provisioned in a step-by-step approach. It requires more programming knowledge than its counterpart and has less reusability, but provides users more control over resource deployment.

Run static application security testing

Static analysis should occur before templates are deployed to test for resource misconfigurations and compliance and security gaps. Many tools are available with prebuilt policy as code standards for common security misconfigurations in IaC, such as overly permissive access, open firewall rules, insecure protocol usage, and plaintext secrets. Any tool used during this process should be selected carefully. Granting a tool access to template files makes the tool a viable target for malicious cyber actors looking

to gain information on an organization's infrastructure or alter templates to enable execution.

Templates should also be deployed in testing environments to ensure the resource definitions function as expected. This is especially important when deploying templates against existing resources as it is possible the resource, such as a database, can be completely overwritten by the new deployment.

D3FEND Tactic	Countermeasure
Application Hardening	Application Configuration Hardening [D3-ACH]
System Mapping	System Vulnerability Assessment [D3-SYSVA]

Enable version control

Version control should be enabled on template files to make tracking changes easier. If a deployment needs to be reverted back to a previous working state, users can use the last working version without the need to create a new template from scratch. Version control also creates auditable records that hold administrators accountable for modifications to infrastructure resources. Separation of duty can be achieved with some version control environments by dividing those who can propose template changes from those who can accept template changes, and ensuring no single user has the ability to perform both. This prevents any one person from being in a position to introduce fraudulent or malicious code without detection.

Integrate CI/CD

Existing CI/CD pipelines can be leveraged for IaC templates, allowing automated deployments for template resources as they are created or updated. This will ensure templates go through the proper security testing phases and stop the deployment if any issues are discovered. Refer to the joint CSI: Defending Continuous Integration/Continuous Delivery (CI/CD) Environments for more information on CI/CD best practices.

After deployment

Perform dynamic application security testing

Resources created from an IaC deployment should go through dynamic testing to ensure they are functioning as expected and all identified threat vectors are properly addressed.

D3FEND Tactic	Countermeasure
File Analysis	Dynamic Analysis [D3-DA]

Enable access control

Access control should be enabled on IaC template files themselves, restricting who is allowed to make changes to IaC code. Users who are restricted from manually deploying or modifying specific resources should also be restricted from modifying IaC templates. Access to templates should be tightly controlled using methods such as multifactor authentication or two-person integrity.

D3FEND Tactic	Countermeasure
Credential Hardening	User Account Permissions [D3-UAP]

Avoid manual changes

Once an IaC template is deployed, manual updates to resources should be avoided to prevent resource drift. Updates should be made in template files, tested, and redeployed through the respective IaC solution.

Enable continuous logging and monitoring

Logging should be enabled on resources deployed through IaC templates and continuously monitored for unauthorized changes. For more information on cloud logging best practices, refer to the NSA CSI: Manage Cloud Logs for Effective Threat Hunting.

D3FEND Tactic	Countermeasure
User Behavior Analysis	Local Account Monitoring [D3-LAM]

Best practices

Organizations are increasingly using IaC to automate the secure deployment of resources to cloud and on-premises environments. IaC eliminates the need for manual deployments, which often lead to resource misconfigurations and security gaps. IaC also provides protections against *ghost* assets, promotes resource standardization, and eases disaster recovery efforts.

Before deploying IaC templates, the following best practices should be considered:

- Create a threat model to identify attack vectors and mitigations that can be put into place to reduce the likelihood of compromise.
- Decide which organizational rules are best encoded as declarative or imperative.
- Run static analysis against templates to test for misconfigurations and security gaps.
- Enable version control on IaC templates.
- Integrate template deployment into existing CI/CD pipelines.

After deploying IaC templates, the following best practices should be considered:

- Perform dynamic analysis to ensure resources deployed correctly and threat vectors are properly addressed.
- Enable access control on IaC templates.
- Avoid making manual changes to resources deployed through IaC.
- Enable continuous logging and monitoring to detect unauthorized changes.
- Audit changes to IaC templates.

For guidance on creating templates and the deployment process, refer to specific vendor documentation.

Further guidance

Supplementary NSA guidance on ensuring network environments are secure and defendable is available at NSA Cybersecurity Advisories & Guidance. Those of particular relevance are:

- Mitigating Cloud Vulnerabilities
- Top 10 Mitigation Strategies
- Advancing Zero Trust Maturity throughout the User Pillar
- Network Infrastructure Security Guide
- Identity and Access Management Recommended Best Practices for Administrators

Works cited

[1] The Open Worldwide Application Security Project (OWASP). Infrastructure as Code Cheat Sheet. 2021. https://cheatsheetseries.owasp.org/cheatsheets/Infrastructure_as_Code_Security_Cheat_Sheet.html

[2] V. Gazdag. A Guide to Improving Security Through Infrastructure-as-Code. 2022. https://research.nccgroup.com/2022/09/19/a-guide-to-improving-security-through-infrastructure-as-code/
[3] The MITRE Corporation. MITRE ATT&CK. 2024. https://attack.mitre.org
[4] The MITRE Corporation. MITRE D3FEND. 2023. https://d3fend.mitre.org

Disclaimer of endorsement

Trademarks

Purpose

This document was developed in furtherance of NSA's cybersecurity missions, including its responsibilities to identify and disseminate threats to National Security Systems, Department of Defense, and Defense Industrial Base information systems, and to develop and issue cybersecurity specifications and mitigations. This information may be shared broadly to reach all appropriate stakeholders.

Contact

Cybersecurity Report Feedback: CybersecurityReports@nsa.gov
General Cybersecurity Inquiries: Cybersecurity_Requests@nsa.gov
Defense Industrial Base Inquiries and Cybersecurity Services: DIB_Defense@cyber.nsa.gov
Media Inquiries / Press Desk: NSA: 443-634-0721, MediaRelations@nsa.gov