Dataweavers

Top 10 Checklist:

Proven Scalable Sitecore Architecture on Azure

www.dataweavers.com

- [ ] **01.** Globally resilient
- [ ] **02.** Blue Green Deployments
- [ ] **03.** Separation of Code, Config, and Infrastructure
- [ ] **04.** Well planned Branching and Release Strategy
- [ ] **05.** Content Delivery Network (CDN)
- [ ] **06.** Frontline Defence Strategies
- [ ] **07.** Configuration Compliance and Scanning
- [ ] **08.** Sitecore Security Hardening
- [ ] **09.** Coding Standards
- [ ] **10.** Changes through Code IAC

01.

Globally resilient

For High Availability (HA) and Disaster Recovery (DR) coverage, ensuring an 'always on' platform without downtime is vital, you need to consider a multi-region deployment, using Azure for example this is possible and enabled by distributing at a minimum the Content Delivery nodes Or Rendering Host (in the case of headless), replicated SQL and managing traffic with Azure Front Door. It is also good practice to run this in an Active pattern (70/30 or 50/50 typically), whereby you are always ensuring both regions are fully operational, and able to serve traffic.

02.

Blue Green Deployments

Your DevOps release process should incorporate blue-green deployments, which allows an updated version to be deployed without disturbing the current website experience. Validation and subsequent cutover are feasible in Azure using Slots in App Services. Coupled with the globally resilient pattern across regions, you can ensure no downtime with careful design

03.

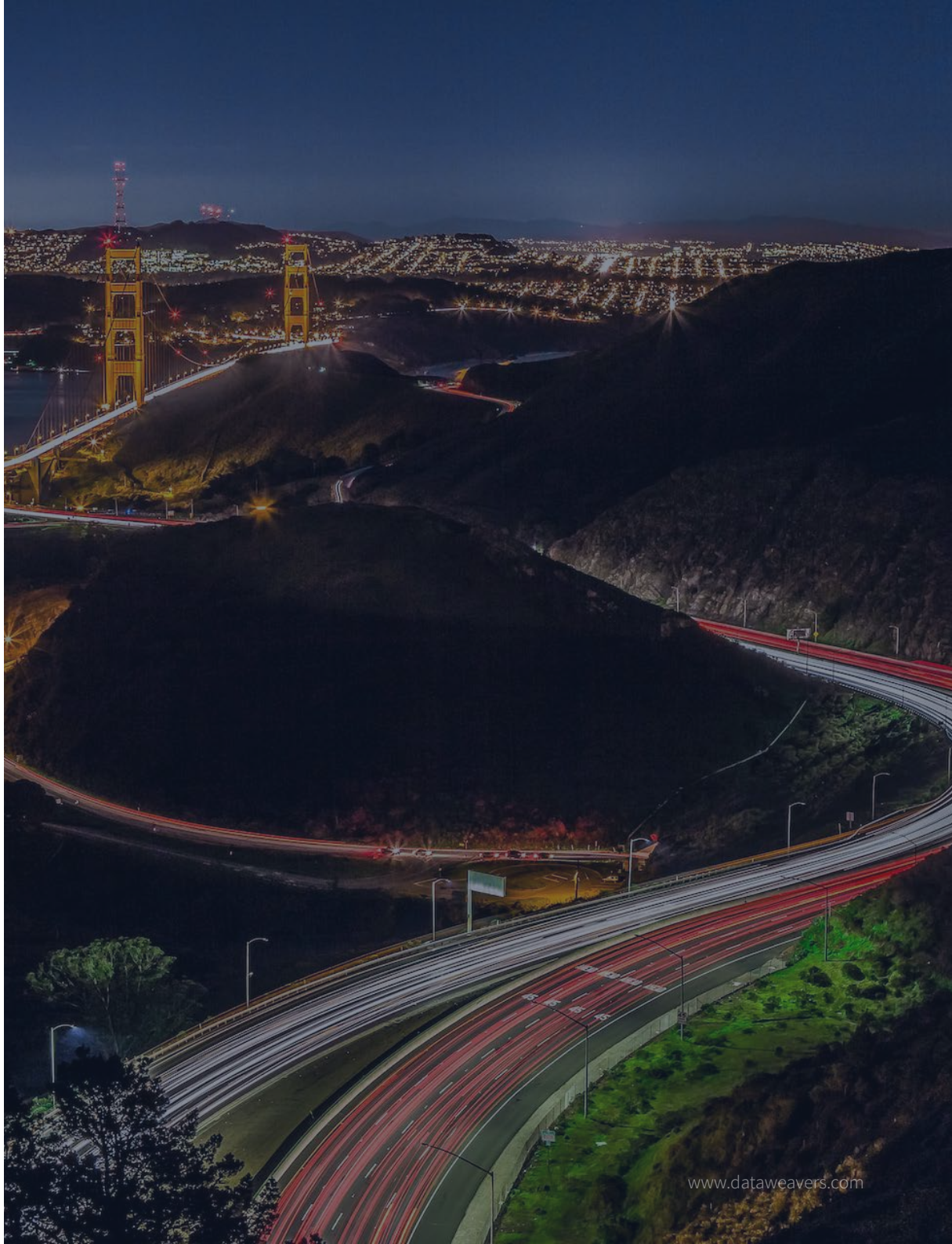
Separation of Code, Config, and Infrastructure

A common issue observed in most environments we review is that the Sitecore application code is tightly coupled with the environment configuration. This often results in infrastructure-related elements being included inappropriately. It is critical to separate the Infrastructure as Code (IAC), the core Application Deployment, the Solution Code, and the core Configuration Management. By managing all core configurations through code and pipelines, you enhance the consistency and efficiency of your deployment process across all environments, from development through to production. This separation not only reduces complexity but also improves the scalability and manageability of your architecture.

04.

Well planned Branching and Release Strategy

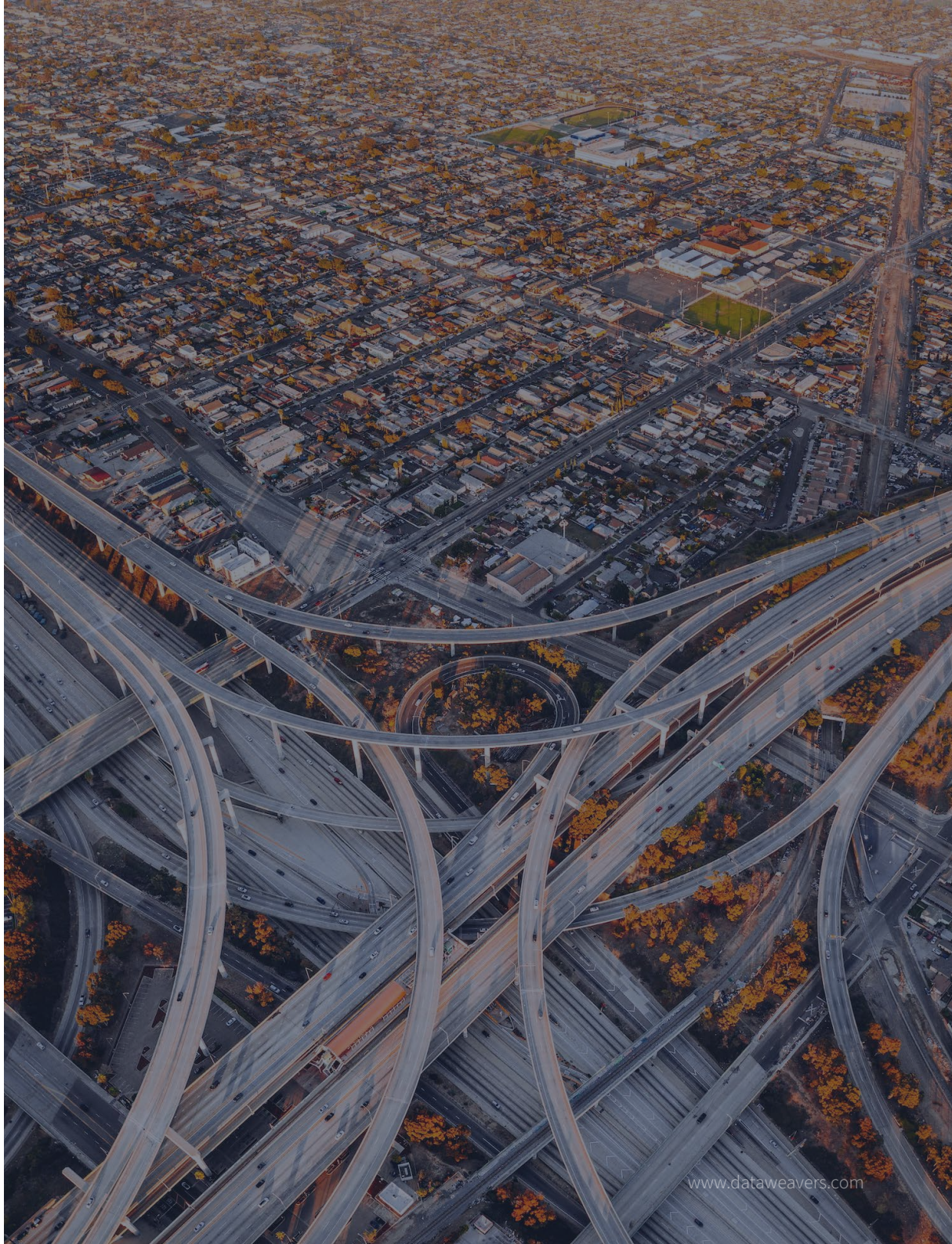
Managing your code repositories can be accomplished using various methodologies, such as Git flow, GitHub flow, or GitLab flow, among others. In addition to the essential recommendation of separating your code, configuration, and infrastructure to maximize flexibility, it is equally important to carefully devise a branching strategy that suits your specific architectural and business requirements. Often, there is a significant gap in planning for the post go-live phase of the application lifecycle. A crucial aspect of a successful operating environment is the ability to efficiently deploy hotfixes. These hotfixes, which need to be evaluated and released, often occur independently of your regular development rhythm. Hotfix deployments are not only essential for addressing critical Business As Usual (BAU) updates but should also be considered for Sitecore platform hotfixes and Security Updates, which are increasingly incorporated into full Cumulative Updates. Neglecting this aspect in your architectural considerations may lead to a situation where you must deploy untested code to production or face difficulties in resolving critical issues. At Dataweavers, we provide guidance to our customers in this area, drawing from proven practices and assisting them in expanding their current capabilities.



05.

Content Delivery Network (CDN)

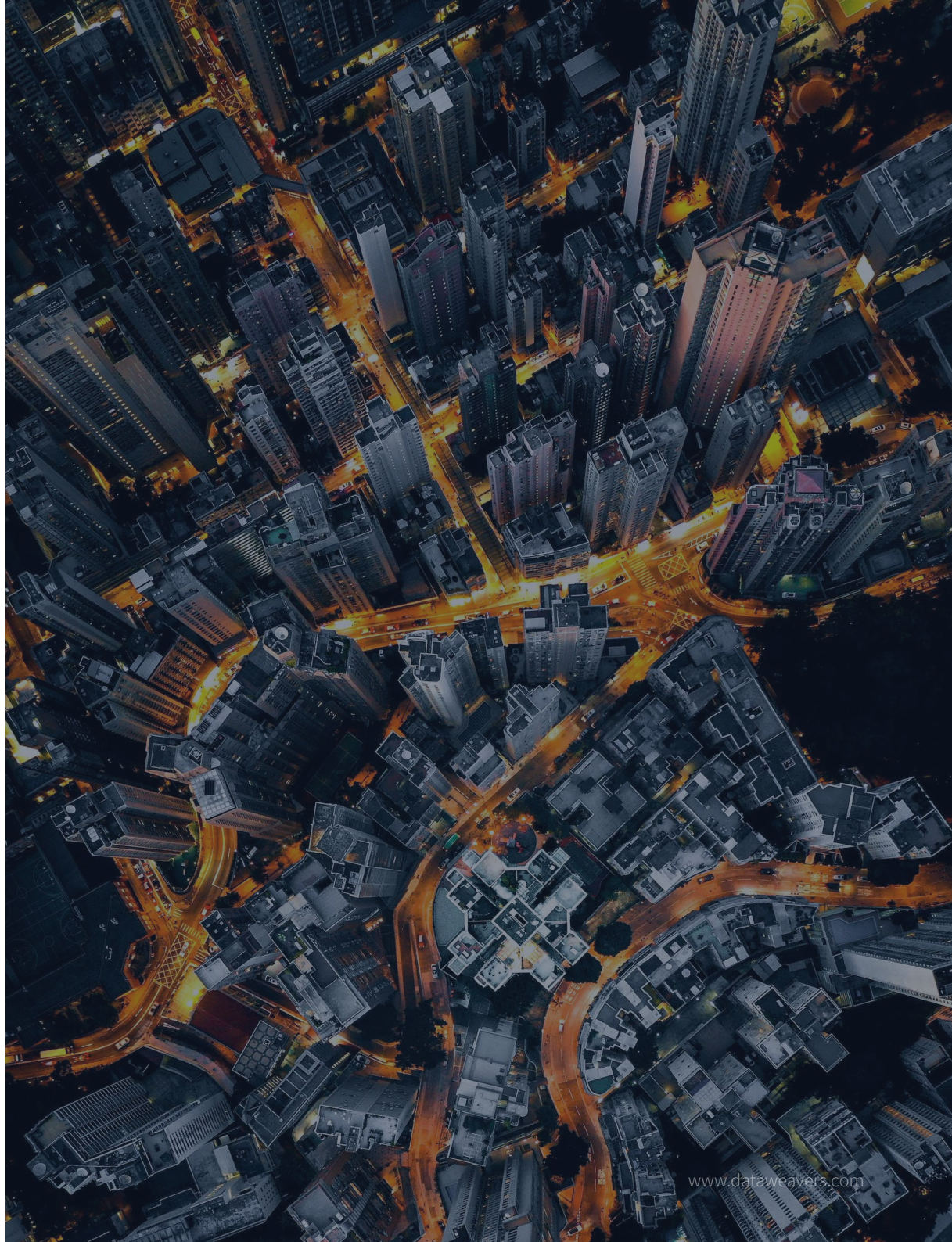
Performance tuning is a multifaceted process, one key element of which is the effective deployment of a CDN platform. Incorporating a CDN into your architecture is paramount, as it significantly reduces latency in asset delivery to client devices. A meticulously configured CDN that takes into consideration Sitecore's unique characteristics, the type of solution, and the geographical location of your users can drastically enhance your site's performance. It is also vital to consider advanced features such as dynamic image resizing. We rely on Cloudflare as a key component of our robust architecture, delivering a highly automated CDN solution to our clients.



06.

Frontline Defence Strategies

Establishing a finely tuned Web Application Firewall (WAF) layer, that at the very least encompasses protection against the OWASP Top 10 vulnerabilities, should be your fundamental baseline. WAF rules require diligent management, with particular attention given to Sitecore vulnerabilities, ensuring secure Content Management (CM) endpoints. A globally scaled DDoS protection mechanism and bot protection should also be integral parts of your security measures and need thorough evaluation to ensure they meet your protection requirements.



07.

Configuration Compliance and Scanning

Using a security configuration scanning platform, purpose-built for the technology you are deploying, can significantly enhance your security posture. A consistent approach to review and apply recommendations is crucial and should be productised. In Azure, we strongly recommend Microsoft Defender for Cloud, a cloud-native application protection platform (CNAPP). This tool not only feeds into Security Centre, enabling you to natively identify risks in your architecture, but it also promotes compliance with your set standards and allows you to evaluate your system against Microsoft's best practices. Moreover, it provides detailed threat information, adding an extra layer of security insight.

08.

Sitecore Security Hardening

Sitecore Security Hardening: Your Sitecore configuration should be validated against Sitecore's Security Hardening principles. These key aspects should be seamlessly integrated into your architecture, as they effectively limit known attack vectors. This ensures your deployment adheres to the recommended best practices, enhancing both the security and the integrity of your system.

09.

Coding Standards

To ensure a secure Sitecore architecture, it is vital to apply recognized coding standards, such as the Sitecore Helix principles. Adhering to Helix provides multiple benefits, including modular and scalable architecture, consistent development practices, improved collaboration and team efficiency, enhanced testability and quality assurance, and access to a supportive community for continuous improvement. By following Helix, you can strengthen the security of your Sitecore architecture and ensure a robust and maintainable solution. Incorporating Static Application Security Testing (SAST) tooling, further strengthens security. SAST tools analyse the source code to identify potential vulnerabilities and design flaws. Integrating SAST tooling into your development process enables early detection of security issues and ensures consistent security checks during the build process.
By combining coding standards with SAST tooling, you can significantly enhance the security of your Sitecore architecture, leading to less vulnerable and better-performing code. We have found that a well configured SonarQube rule set provides significant benefits in this area.

10.

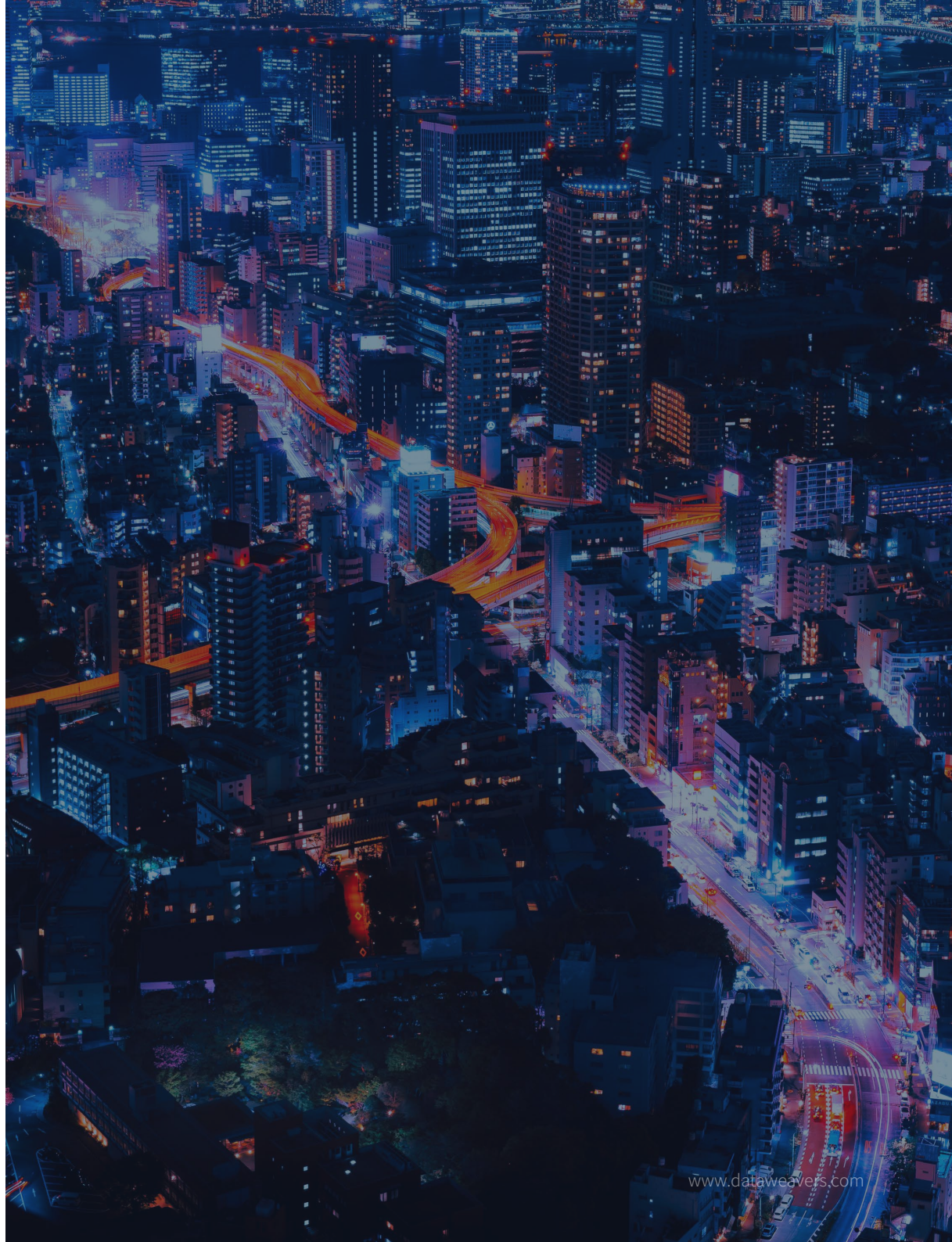
Changes through Code IAC

Extending the separation concept is how you make infrastructure level changes. A conscious approach to ensure Infrastructure as Code Principals always apply is fundamental. It takes work, but an IAC template that is only used at initial deployment, and then rapidly becomes out of date, poses significant risk to your architecture Agility. Having your IAC templates and Pipelines fully functional and current is a critical aspect of a well-oiled architecture.

Dataweavers has fully productised these 10 points (and more) in our WebOps platform, and we fully remove the initial and ongoing operational concerns of you having to create and maintain the best practice Architecture.

Sitecore also provides a starting point for **point 10 Infrastructure As Code (IAC)**. The default Sitecore ARM templates will need consideration for your specific scaling, monitoring, cost, usage, and performance needs, but provide a good start. They are available in the following GitHub repo:

GitHub - Sitecore/Sitecore-Azure-Quickstart-Templates: This repo contains all currently available Azure Resource Manager templates for Sitecore.



Contact us

hello@dataweavers.com

www.dataweavers.com/contact

Or call one of our team

Australia ..... +61 483 901 121

United Kingdom ..... +44 7400 374066

United States ..... +1 (888) 817-3688