WHITE PAPER

IoT Cloud Deployments using IaC Methodologies

Cloud-based IoT solutions require quick, easy and error-free deployments on demand. This is essential for implementing DevOps processes for IoT solutions.

Infrastructure as Code (IaC) enables provisioning and managing IT infrastructure using source code, rather than manual processes or standard documented procedures.

Using tools from AWS® and Azure®, IoT cloud solutions can be easily and rapidly deployed, in an extremely reliable, repeatable, error-free manner.

THINXTREAM®

Introduction

Legacy infrastructure management methodologies involved manual documented processes, brittle scripts, and a few graphical user interface (GUI) based tools. Today, the equivalent cloud methodologies involve manual provisioning of services from a portal, configuring various service parameters, and interconnecting them to achieve the solution objectives.

These methodologies work, but with limitations, and are time-consuming. Issues in terms of delays, configuration errors, and negotiating the approval chain for any requirements from the development teams get magnified with the enormous scale of Internet of Things (IoT) deployments – where massive scale message handling, container-based deployments, micro-services, and big data analytics come into play in the cloud infrastructure.

With the perpetual and growing needs on scaling infrastructure, requirements of transient and on-demand infrastructure, and complex and integrated systems and services, businesses require better ways of managing the cloud infrastructure for IoT solutions. This becomes extremely important where DevOps is the norm and closely integrated and numerous services require fast, error-free deployments for development, testing, staging and production.This white paper shares Thinxtream's experience in addressing these issues with Infrastructure as Code (IaC) using tools from Azure® and AWS®.

IoT Cloud Challenges

Deployment of the next-generation cloud infrastructure for IoT solutions requires a change in the way IT departments and systems work, but the main challenges remain similar.

- Identify and develop the change required to address the problem, and incorporate into the main codebase.
- Deploy the change to a staging environment to ensure that the change is safe and correct. Run the required tests.
- Ensure the deployment configuration is consistent and free of discrepancies.
- Deploy and orchestrate the change to the production infrastructure.
- Monitor the health and performance of the newly-deployed infrastructure.
- Track the configuration change history, including the change requester, approver, and also the deployment history.
- Ensure that the above steps are done in a fast, efficient, reliable and secure manner.

What is IaC?

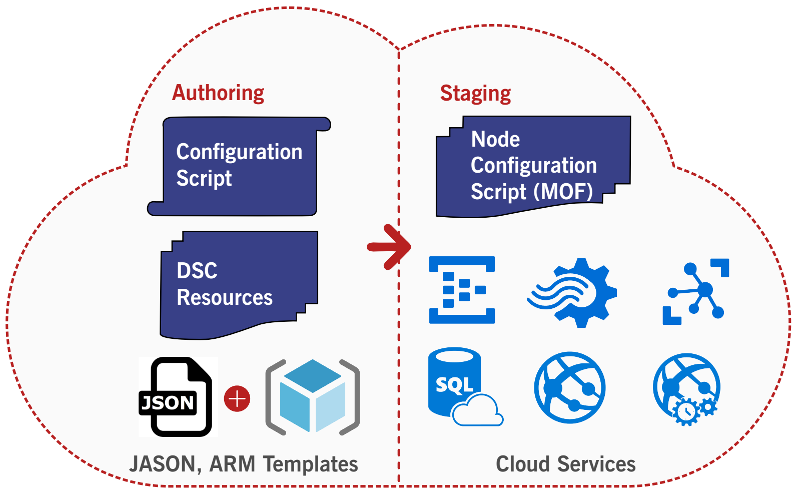

Figure 1. Azure Automation with DCS & ARM

Infrastructure as Code (IaC) enables provisioning and managing IT infrastructure using source code, rather than manual processes or standard documented procedures. All the required infrastructure is scripted, along with their configurations, dependencies, external input parameters and interconnections, and managed as a source code.

Microsoft® Azure® uses Azure Resource Manager (ARM) templates and Azure Automation DSC extensions, while AWS® achieves this using AWS CloudFormation® and AWS CodeDeploy®. Similarly, in the case of Google™, Google Cloud Deployment Manager provides the IaC infrastructure.

Azure Resource Manager templates consist of JSON and expressions that can be used to define a deployment. Azure also provides Desired State Configuration for bootstrapping Virtual

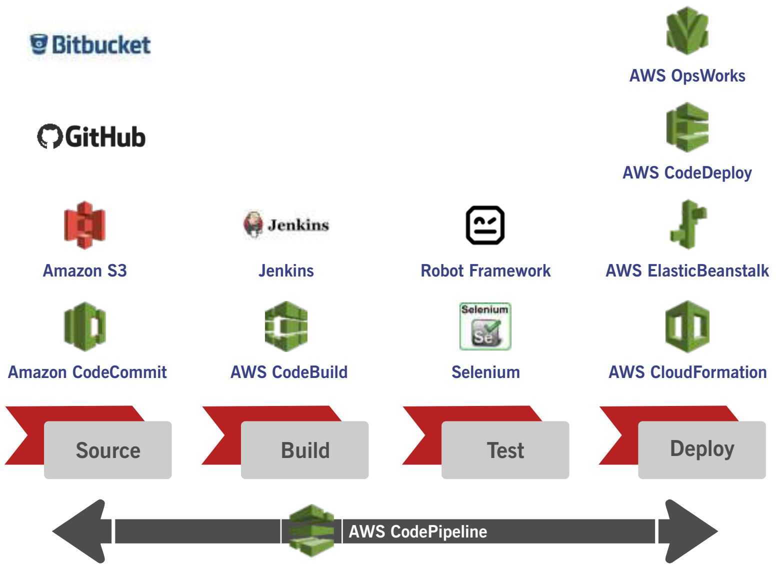

Figure 2. AWS Code Pipeline

Machines (VMs) to enable ongoing management of VM configuration and integrating with other monitoring tools like Azure Monitor. The ARM templates and DSC extensions together enable the building of a robust consistent and reliable IaC based methodology.

AWS CloudFormation is similar to ARM Templates. It uses a simple text file (either YAML or JSON) to describe and provision all the cloud infrastructure resources across regions, in an automated and secure manner. The template can be validated for syntax and semantic errors using AWS CloudFormation console or AWS Command Line Interface (CLI). Further, AWS CodeDeploy can be used to automate application deployments to Amazon EC2® instances, or deploy server-less AWS Lambda® Functions, or applications using Amazon S3® buckets, GitHub®, GitLab®or Bitbucket® repositories.

Some of the other solutions, which can be used to build an IaC infrastructure are Ansible®, Chef™, Puppet™, Terraform®, Foreman and Docker®. Some of these can be used for Virtual Machine configuration management on Azure and AWS clouds, but they cannot efficiently handle the ever increasing cloud services and solutions from Azure and AWS.

Benefits of Using IaC

You can use IaC methodologies to deploy IoT solutions. For instance, in the case of Azure IoT Hub, the service parameter definitions and the various interconnections between IoT Hub, Messaging Queues, and Storage Services, Web Jobs, and Analytics cloud services can be defined in a parameterized JSON document. This method provides highly reusable deployment templates, which can be used to vary the cloud service configurations for each deployment. During development, the solution generates only a few messages, so the development team can use the basic service level with minimal, non-redundant storage for their development environment. After development, the production deployment requires larger message ingestion support, larger storage capacity and global redundancy. Both of these deployments can be achieved using the same JSON template document, by just varying the input parameters.

Faster Deployments using Simple Methods

Azure Resource Manager or AWS CloudFormation templates can be used to define Virtual Machines, Pre-configured Databases, Network Infrastructure, Load Balancers, IoT Services, Analytics Services, Docker based Micro-services with Kubernetes orchestration and several other inter-connected cloud services. This can be easily done for development, staging and production environments, using the same script with different parameters to suit the deployment requirements. The infrastructure can be grouped using Resource Groups, deployed to any region across the globe, and even be used for Backup and Disaster Recovery.

Azure deployments can be done using Visual Studio® Team Services (VSTS) from GitHub, or any other source code repository. Similarly, for AWS, source code is accessed from GitHub, or other source code repositories, resources from S3, and further, a code pipeline integration is created to build, test, and deploy the full solution using AWS CloudFormation and AWS CodeDeploy.

Achieve Repeatable Configuration Consistency

The IaC templates for Azure and AWS standardize the setting up of the cloud solution, which minimizes errors and configuration differences. They achieve complete consistency across deployments, in a repeated manner, without creating any compatibility issues with the cloud infrastructure. Along with Azure DSC or AWS CodeDeploy, which can ensure bringing up a VM to the required configuration level, by installing the required applications or OS components, the cloud IaC infrastructure can help teams get their solutions up and running in quick time.

Effective Risk Mitigation and Management

IaC templates and configuration customization services provide a form of documentation of the infrastructure for easy reference and updating. Knowledge about the cloud setup is transferred into the code, thus mitigating risks of people change. Further, since the IaC source code files are configuration controlled, changes are effectively tracked and any deployment issues can be easily traced back or reversed, if necessary. Gated check-ins ensure code quality and integrity by triggering unit tests before changes go into the master branch.

Efficient Software Development

IaC based deployments can be done in multiple stages, which makes the software development life cycle more efficient. Developers, QA, Staging and Production can have their own deployments, and also re-deploy them as and when needed, without any support from IT teams. The IaC approach using ARM/DSC/VSTS or CloudFormation/CodeDeploy enables continuous delivery, continuous integration and continuous deployment minimizing human errors during the entire development cycle.

The IaC scripts can also enable efficient cost-management by shutting down unused environments, cleanly and completely.

Conclusion

Deployment requirements for next-generation cloud infrastructure for IoT solutions require a change in the way IT departments and systems work. IaC enables provisioning and managing IT infrastructure using source code, rather than manual processes or standard documented procedures. It enables faster deployments using simple methods, repeatable and consistent configuration, effective risk mitigation and management, and efficient software development.

While there are several alternatives to implementing IaC, ARM templates, DSC Extensions and, DSC Automation, and AWS CloudFormation and AWS CodeDeploy are the best ways to achieve this on Azure and AWS respectively. As an IoT services provider, Thinxtream has extensive expertise in leveraging IaC using Azure and AWS. By leveraging the IoT expertise built over a decade, Thinxtream ensures cost-effective, quality and timely delivery of IoT solutions.

References

- https://techbeacon.com/infrastructure-code-engine-heart-devops
- https://searchitoperations.techtarget.com/news/450280797/Infrastructure-as-code-tops-ITs-DevOps-challenges
- https://puppet.com/blog/what-is-infrastructure-as-code
- https://www.cio.com/article/3017722/infrastructure/what-is-infrastructure-as-code-and-why-should-you-embrace-it.html
- https://stelligent.com/2017/06/29/devops-benefits-of-infrastructure-as-code/

Thinxtream Technologies is a global software company with a portfolio of innovative software platforms, components, solutions, patents, competences and services for Internet of Things (IoT) across several industry verticals and applications, successfully enabling leading customers, including Fortune 500 companies, meet their application, product and business goals.

Interested in learning more? For more information contact:

Thinxtream Technologies Pte. Ltd.
220 Orchard Road #05-01
Midpoint Orchard
SINGAPORE 238852
Phone: +65 66358625
Email: info@thinxtream.com

Thinxtream Technologies, Inc.
10260 SW Greenburg Road
Suite 400 Portland, OR 97223
U.S.A.
Phone: +1 971 230-0729
Email: info@thinxtream.com

www.thinxtream.com

LinkedIn/thinxtream

TT-WP-004-3-1220

THINXTREAM®