A DevOps Reference Architecture for Multi-Cloud IoT Applications Deployment

Georges Bou Ghantous

Doctor of Philosophy (C02029)

Software Engineering

University of Technology Sydney

Faculty of Engineering and Information Technology

School of Computer Science

Certificate of Authorship

I, Georges Bou Ghantous, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy in Software Engineering in the Faculty of Engineering and IT, School of Computer Science at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note:
Signature removed prior to publication.

Date: 22/04/2020

Acknowledgments

I have been privileged to have Dr Asif Gill and Dr Farookh Hussain as my supervisors. I want to express my sincere gratitude to my supervisor Dr Asif Gill for providing me with an opportunity to work with him on this research project. His highly valuable support, coaching, encouragement, quick feedback, and guidance helped me to finish this research project. Dr Asif Gill was always there to provide support in meetings, phone conferences, and emails. His valuable understanding in the field of agile and software engineering as an academic, researcher, and practitioner provided a wealth of knowledge that I used in this research project.

This study would not have been possible without the support of my family. I am very grateful to my parents, Salim Bou Ghantous and Laure Mikhail, for their precious love, blessings, support, encouragement, sacrifice, and care since my childhood. I am thankful for the support of my siblings. Their genuine care gave me confidence and peace of mind during my busy life. I am blessed to be a member of such a loving, caring, and supportive family.

I want to express my sincere thanks to all of my colleagues and friends for their valuable feedback and support—especially my dearest friends, Charbel Lahoud and John Mikhael (late).

I also wish to thank the Australian Government for its support and HECS-HELP for providing the funding for my research project for the length of my PhD study period.

I am thankful to all of the reviewers for valuable feedback and comments.

Thank you all.

Research Contributions and Publications

During this PhD research project, I collaborated with my supervisor and other colleagues. I published the components of this research work (DRA) in several rigorously reviewed international conference papers and scientific journals. The papers' publications were an opportunity to present my work for review before including it in this thesis. Appendix F presents a list of the publications that have been included in this thesis.

Contents

List of Tables

List of Figures

List of Equations

List of Abbreviations

AAF	Average and Above Frequency
AAP	Average and Above Percentage
API	Application Programming Interface
App	Software Application
CD	Continuous Deployment
CI	Continuous Integration
DB	Database
DRA	DevOps Reference Architecture
DSR	Design Science Research
IaaS	Infrastructure as a Service
IoT	Internet of Things
MQTT	Message Queuing Telemetry Transport
PaaS	Platform as a Service
PC	DevOps Practice
POC	Proof of Concept
QEM	Qualitative Evaluator Matrix
QIM	Quantitative Indicator Matrix
QI	Quantitative Indicator
RDF	Resource Description Framework
Retro-QA	Retrospective Quality Assurance
RFID	Radio Frequency Identification
RPIB	Raspberry Pi Model 3 B
RQ	Research Question
SaaS	Software as a Service
SLR	Systematic Literature Review
SSH	Secure Shell
UTS	University of Technology Sydney
WAN	Wide Area Network
WSN	Wide Sensor Network
GPIO	General-purpose input/output
FEIT	Faculty of Engineering and Information Technology

Glossary

Abstraction	Abstraction refers to a logical view of entities such as objects, elements and services.
Agile	Agile is an iterative software development methodology that solves the complexity of the software project by adopting an iterative approach to 'revisit' the development process (analysis, planning, architecture, design, develop, test and deploy) and hasten the release or delivery of products to customers.
Continuous Integration (CI) Broker	A CI broker is a DevOps cloud-based tool that hosts the deployment configuration for the software applications (e.g. IoT-applications) deployment to multi-cloud in a DRA architectural model.
DevOps	DevOps is a set of software development practices that combine software development (Dev) and information technology operations (Ops) to improve agile software development.
DevOps Pipeline	A pipeline is a set of integrated DevOps tools that enable an automated software deployment process.
Framework	A framework is a set of development elements and components that are combined to produce a tailored process/method.
Internet of Things (IoT)	IoT refers to a network of physical objects that feature an IP address for internet connectivity, and the communication that occurs between these objects and other internet-enabled devices and systems.
JavaScript Object Notation (JSON)	JSON is an open-standard file format that uses human-readable text to transmit data objects consisting of attribute-value pairs and array data types (or any other serializable value).
Multi-Cloud	Multi-cloud is the integration of several cloud computing services in a single heterogeneous architecture.
NoSQL Database (DB)	A No SQL DB stores data in a document format such as JavaScript Object Notation.
Process/Method	'Process' and 'method' are used interchangeably in this thesis (although it is acknowledged that these terms are used differently in other areas of software engineering).
Reference Architecture	Reference architecture in the field of software architecture or enterprise architecture provides a template solution for architecture for a particular domain.
Vendor Lock-In	Vendor lock-in is a situation in which a customer using a cloud product or service cannot easily transition to a competitor's cloud product or service. Vendor lock-in may occur in several cases—for instance, when a cloud in the multi-cloud system hosts the deployment configurations of the software application and when a cloud in the multi-cloud system hosts the database for the application data.
Waterfall	Waterfall methodology aligns the software development process (analysis, planning, architecture, design, develop, test, and deploy) in sequential order.

Abstract

DevOps originated in the context of agile software development, which seems an appropriate approach to enable the continuous delivery and deployment of a software application in small releases. There is growing interest among organisations in adopting the DevOps approach and a multi-cloud environment for IoT (Internet of Things) application deployment. However, the challenge is how to apply DevOps when a multi-cloud heterogeneous environment is required for IoT application deployment. To address this vital research need, this thesis applies a design science research (DSR) method. It develops the DevOps reference architecture (DRA) framework to automate IoT applications deployment to the heterogeneous multi-cloud environment. The DRA is a cloud-enabled framework that mainly focuses on the deployment part of the integrated agile–DevOps methodology. Using a DSR method, the DRA has been incrementally developed by the iterative application of build, review, and adjust research activities. The DRA is intended for use by software organisations, coaches, managers, engineers, developers, and consultants as comprehensive reference architecture for deploying IoT applications to a multi-cloud environment using the DevOps approach.

The DRA has three main components: framework characteristics, framework architecture, and framework composition. Framework characteristics incorporate nine main elements arranged into three categories: foundation (abstraction, human factor, infrastructure), core (process, tools, product), and extended (business value, rules, legal). Framework characteristics provide the building blocks necessary to create a reference architecture design using the DevOps approach and cloud infrastructure. Framework architecture is composed of five models: contextual, conceptual, logical, physical, and operational. Framework architecture is the blueprint used in the framework composition to create DevOps pipeline instances that enable IoT application deployment to the multi-cloud environment. The DRA framework composition includes three components: resources (architecture design, software, and hardware), configuration (pipeline, IoT application, IoT network), and output (DRA reference architecture, DRAv1.0 instance, DRAv2.0 instance). The framework provides implementation instructions and an evaluation template to implement and evaluate DRAv1.0 (single cloud) and DRA v2.0 (multi-cloud) instances in different organisational contexts.

The proposed DRA framework is evaluated using an empirical evaluation composed of four iterations: industry case study, research case study, teaching case study, and industry field surveys. The results of this thesis indicate that the proposed DRA framework can be considered reasonable for the successful adoption of the DevOps approach for IoT application deployment to the multi-cloud environment. The evaluation results indicate that the DRA framework is generic and can be used in different organisational contexts and technology stacks to establish a cloud-based deployment architecture that is suitable for IoT applications.