The DevOps Reference Architecture Evaluation

A Design Science Research Case Study

Georges Bou Ghantous
University of Technology Sydney
Sydney, Australia
Georges.BouGhantous-1@uts.edu.au

Asif Gill
University of Technology Sydney
Sydney, Australia
Asif.Gill@uts.edu.au

***Abstract*— There is a growing interest to adopt vendor-driven DevOps tools in organizations. However, it is not clear which tools to use in a reference architecture which enables the deployment of the emerging IoT applications to multi-cloud environments. A research-based and vendor-neutral DevOps reference architecture (DRA) framework has been developed to address this critical challenge. The DRA framework can be utilized to architect and implement the DevOps environment that enables automation and continuous integration of software applications deployment to multi-cloud. This paper confers and discusses the evaluation outcomes of the DRA framework at the DigiSAS research Lab. The evaluation outcomes present practical evidence about the applicability of the DRA framework. The evaluation results also indicate that the DRA framework provides general knowledge-base to researchers and practitioners about the adoption DevOps approach in reference architecture design for deploying IoT-applications to multi-cloud environments.**

***Keywords*— Design Science Research, DevOps Reference Architecture, Empirical Software Evaluation, IoT Application Deployment, Multi-Cloud Automated Deployment**

I. Introduction

DevOps approach promises to enable continuous integration, continuous deployment and fast, automated delivery of software applications in small releases [3], [11], [16]. The practices of current software development and deployment methods, including agile, led to the emergence of integrated Agile-DevOps automation paradigm [3], [15]. DevOps provides practices to bridge the gap between 'Dev' and 'Ops' and improve team collaboration [1], [2] in the overall context of agile software development [15], [20].

The Internet of Things (IoT) is an important emerging technology that incorporates the connection and communication between physical devices (IoT-sensors) and virtual software [9], [10], [23]. IoT software applications require continuous integration, automation and real-time monitoring. The mentioned concepts could be achieved by applying DevOps practices and using DevOps tools [1], [9]. Hence, DevOps approach could be adopted for IoT-applications deployment to multi-cloud [1], [17].

This research presents a research-based and practical DevOps reference architecture (DRA) [1]. The DRA architectural design is founded on five models: 1) contextual, 2) conceptual, 3) logical, 4) physical, 5) operational. The DRA has been constructed using the guidelines of the design science research (DSR) method [5]. The main contribution and scope of this paper is the evaluation of the DRA framework in DigiSAS research lab using case study template. The results of the evaluation are discussed to determine the applicability, reusability and usefulness of the DRA framework in the research lab context.

II. Research Background and Related Work

In agile software development context, the objective of DevOps approach adoption is to improve cooperation and between Dev and Ops [2], [11]. DevOps offers a set of well-known practices that provide supportive guidelines for a broader perspective to develop and deploy software applications to the cloud [2], [19], [20]. IoT is increasingly receiving attention in the IT industry [21] and would benefit for DevOps to facilitate the human–sensors interactions using software applications in a secure environment [9], [18]. The IoT value for organizations exists in the automated operations of IoT applications. Similar to DevOps, IoT applications are complex and involve real-time operations with IoT devices. The performance of IoT applications is determined by measuring the connection protocol latency (MQTT, RSSI, NFC, Wi-Fi, and mobile [33]). The performance of IoT-applications is also deduced by managing IoT data stored either in conventional SQL tables or by using NoSQL database [24], [34].

The cloud offers potential solutions that may aid in overcoming the challenges presented by IoT paradigm [22], [25], and [26]. Organizations and researchers can benefit from cloud platforms integrated into the multi-cloud system [27], [31]. However, the multi-cloud system does not promote distributed application deployment. The major obstacle for adopting multi-cloud is vendor lock-in, which prevents harmonious deployment and database integration for the software application [28], [29]. Vendor lock-in may occur when a cloud from the multi-cloud system hosts the deployment configuration or when a cloud hosts the database. The multi-cloud seems to aid in automated software deployment by offering essential services [10]. IoT can benefit from multi-cloud services [24] and techniques that enable portability and interoperability [13], [30], and [32]. DevOps seems to offer the multi-cloud a set of practices and tools that assist automation and continuous integration across the deployment pipeline [1], [12].

The investigation into the context of DevOps, multi-cloud, and IoT indicates that DevOps adoption for IoT application deployment to the multi-cloud lacks contextual guidelines. The related work analysis identified several research gaps as follows:

- Need to automate IoT applications deployment.
- Need to manage connectivity between IoT-applications and sensors.
- Need to avoid deployment vendor lock-in in multi-cloud.
- Need to avoid database vendor lock-in in multi-cloud.

The results of the evaluation in this paper have been used to determine that the new DRA framework may offer a practical solution to the needs listed in the research gaps.

III. DESIGN SCIENCE RESEARCH

The DRA has been created by means of a well-known design science research (DSR) methodology [5]. The DSR method objective is to offer provable contributions of the DRA applicability in real-world settings. The DSR is composed of three primary stages (Fig. 1):

- Stage 1—DSR flows:
  - Research Background and related work analysis.
  - Related work in publications [1], [2], and [3].
- Stage 2—DSR steps:
  - Problem identification: Initial research into the background and related.
  - Analysis: The analysis results of the research background and related work.
  - Design: The new DRA framework.
  - Development: The new DRA framework.
  - Evaluation: The evaluation of the DRA.
  - Outcome: The evaluation results.
- Stage 3—DSR outcomes:
  - Research problem (section II).
  - Suggested solutions (section II)
  - Design artifact (section IV).
  - Development artifact (section IV).
  - Research Case Study (section VI).
  - Discussion (section VII).

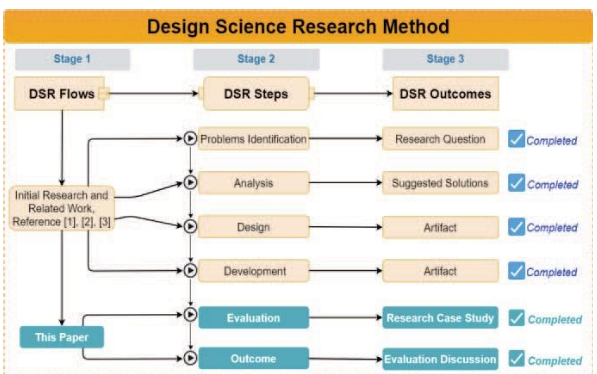

Fig. 1. Design Science Research Method

IV. THE DEVOPS REFERENCE ARCHITECTURE

The DRA is founded on DevOps concepts and practices [2] and on cloud/multi-cloud ecosystem. The DRA design models provide practical solution to support of IoT applications automated deployment to multi-cloud [1], [10], and [12]. The DRA reference architecture design is composed of five models: contextual, conceptual, logical, physical, and operational [1] (please see Appendix for more information about the DRA).

A. DRA Contextual Model

The DRA contextual (Fig. 2) model outlines the relationship between DevOps, Multi-Cloud, and IoT at a higher contextual level. DevOps and Multi-Cloud aim to support IoT-applications deployment [1], [10], [14]. The new concept is the CI-Broker (continuous integration broker). The CI-Broker is a vital mechanism needed to perform several tasks necessary for the IoT-application deployment to multi-cloud. The CI-Broker automates the test/build/deploy operations. The CI-Broker hosts the deployment configurations for the IoT-application to avoid vendor lock-in.

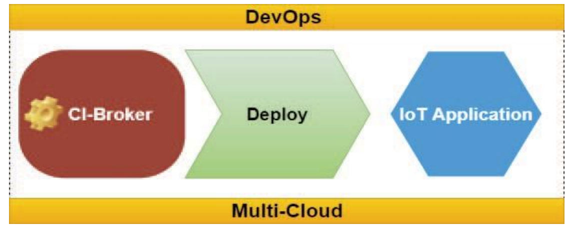

Fig. 2. DRA Contextual Model (Ghantous and Gill 2018)

B. DRA Conceptual Model

The DRA conceptual model (please see Appendix - Case Study Template – DRAv2.0) expands the concepts from the DRA contextual model. The conceptual model introduces the CI broker; an essential utility to prevent vendor lock-in. The CI broker enables continuous integration, branching development and automation (for build, testing, code synchronization). Most importantly, the CI-Broker hosts the deployment configurations for the IoT application, which prevents any of the clouds incorporated in the multi-cloud platform from hosting the IoT application deployment configurations and consequently prevents vendor lock-in.

C. DRA Logical Model

The DRA logical model (please see Appendix - Case Study Template – DRAv2.0) is composed of five components (M1 to M5). The logical model components include the necessary functions to enable DevOps concepts and cloud services integration. The logical model transforms the DevOps practices [2] into features and functions to support the IoT application deployment to the multi-cloud.

D. DRA Physical Model

The DRA physical model, please see Appendix - Case Study Template – DRAv2.0) is a tangible implementation of the logical model. The physical model presents a pseudo-material blueprint of the DRA instances. The DRA instances are the development and deployment pipelines defined by the DRA Operational model.

E. DRA Operational Model

The DRA operational model provides a practical implementation guideline for creating integrated deployment pipelines. The pipeline instances enable the logical model features and functions. The DRA operational model pipeline instances (please see Appendix - Case Study Template – DRAv2.0) are configured using an integrated set of DevOps tools and multi-cloud services [1], [2] that operates as follows:

- The software code is pushed from M1 to M2.
- M2 (CI-Broker) enables distributed deployment of the software application (IoT-application) to M3.
- M2 (CI-Broker) prevents vendor lock-in by hosting the software application deployment configurations.
- M4 model enable real-time monitoring, and communication capabilities.
- M5 manages the IoT-application data collection and storage in NoSQL on separate cloud database.
- M1 monitor the deployment, runt-time, build and testing logs.

V. Evaluation Overview

The DRA has been evaluated using a research case study, which was conducted at the DigiSAS Lab (please see Appendix). The DRA evaluation process overview is illustrated in Fig. 3.

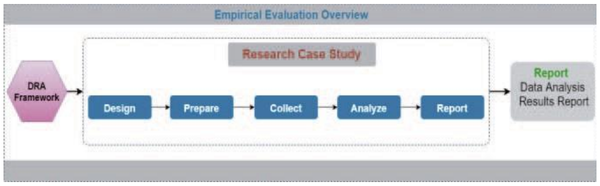

Fig. 3. Empirical Evaluation Overview

A. Case Study Structure

The case study method is commonly used for software engineering oriented DSR research artifact evaluation [6]. Software engineering case studies adopt a positivist view, especially for expressive and theoretical-abstract research [8]. The case study investigation aims to determine if the phenomenon can be replicated and reused in real-world settings. The case study method is composed of five steps:

- **Design**: Plan the case study and identify the objectives.
- **Prepare**: Define the data collection method.
- **Collect**: Outline and explain the case study data collection and data storage.
- **Analyze**: Analyze the collected data using the hypothesis confirmation technique [6]. The hypotheses are the evaluation criteria in Table 1.
- **Report**: Outline the findings of the case study.

B. Evaluation Criteria

The Case Study Template (CST) (please see Appendix) enables the participating organization to test the DRA applicability and provide feedback. The feedback is analyzed using criteria (please see Table 1). The evaluation criteria elements are derived from artifact evaluation and validation criteria in design science research [4], [7].

TABLE I.

Criteria	Evaluation Criteria
	Description
Generalizations	DRA is general in the sense that it is not fixed to one situation or environment.
	DRA instantiable to a class of situations and be used with different technology stacks.
Usefulness	DRA is useful in the organization context.
	DRA can be used as a blueprint for IT projects.
Novelty	DRA offers new knowledge based on DevOps practices.
	DRA offers the new CI broker mechanism
Coverage	DRA provides a sufficient explanation for DRA.
	DRA offers features required for a class of problems.
Reusable	DRA can be replicated for a class of problem situations.
	DRA instances can be configured using various tools with different technology stacks.

[a.] The evaluation criteria are used in the research case study to evaluate the DRA models

VI. Research Case Study

The evaluation of the DRA was conducted in the DigiSAS research lab context using a case study evaluation template (please see Appendix). The research case study steps are explained as follows:

A. Case Study Design

The research case study is organized as follows: (please see Appendix).

- Case study organization context: DigiSAS lab conducts applied practice-based research and development in collaboration with industry partners; working on several software-related projects involving mobile, drone, web, IoT software applications, and the multi-cloud. The partners are from large to small- and medium-sized enterprises (SMEs) and start-ups.
- Need and problem: A multi-cloud environment for software development and deployment that meets the needs of different industry partners. The challenge is how to deploy software applications to multi-cloud.
- Solutions: The DRA seems to address the abovementioned need and problems. The DRA has been explained and used as a guideline framework for setting the DevOps for the multi-cloud.
- Objective: The objective is to evaluate the applicability of the DRA in the research lab environment. The Lab's objective is to have a working DevOps environment for multi-cloud IoT application deployments.
- DRA POC (proof of concept): A demonstration pack was developed to demonstrate the applicability of the DRA framework:
  - Demo Video YouTube video: Link
  - Presentation Slides: Link

B. Prepare

The evaluation was conducted at the DigiSAS Lab using a case study evaluation template (CST) (please see Appendix). Before the formal evaluation, the DRA POC demo was presented to the DigiSAS lab members and industry partners during the quarterly Lab event held on 23/04/2019 at UTS under the supervision of the lab leader. Overall, the lab members and industry partners appreciated the DRA, in particular, the concept of CI-Broker for multi-cloud. The final and formal evaluation was conducted involving the lab leader (LL) on 15/08/2019 at DigiSAS Lab who provided their feedback on the DRA components explained in the CST template (please see Appendix):

C. Collect

The participant provided valuable inputs summarized by the (LL) feedback about the DRA applicability in the research lab context. The total duration of data collection, including demo, presentation, and case study contribution, was approximately 60 minutes. The case study data was stored on CloudStor (please see Appendix). The expert (LL) reviewed the framework design and imparted vital feedback about the DRA models and components with further opportunities for improvements (please see Table 2).

D. Analyze

The case study data collected during the experiment are analyzed in Table 2. The data analysis uses the cross-examination method between (LL)'s feedback and the case study evaluation criteria in Table 1. This analysis aims to connect the hypotheses (evaluation criteria) to the expert's feedback as follows:

TABLE II.

Data Collection and Analysis	
Participant Feedback	***Criteria***
'The output of this research is the DRA artifacts, and the outcome of this research is **new scientific or design knowledge about the DRA itself**. As a research group leader, I reviewed the DRA from the following four perspectives, and my comments are noted below: **Usefulness:** DRA is **applicable and is fit for** setup the DevOps multi-cloud IoT environment for lab research projects**.** **Generalization:** DRA is **general** in the sense that it is not fixed to one situation or environment and can adapt to different situations and **be used with different technology stacks** as appropriate to the situation. Thus DRA is applicable to a class of problem situations and is applicable to several instantiations. **Novelty:** DRA **offers new knowledge**, which has not to be discussed before in the form of complex DevOps for Multi-cloud and IoT. **In particular, the concept of a broker DevOps Cloud in the DRA**. **Explainability:** DRA models seem to provide **sufficient explanation** about the elements and their relationships as a "design knowledge," which can be **used or reused** for a class of a problem addressed in this work**.** My overall feedback is that DRA can be **successfully instantiated for the similar research lab environment needs** for the deployment of IoT applications using multi-cloud. Overall, DRA is fit for purpose; however, the following are some opportunities for further research and development, perhaps new PhD projects'.	Generalizations Usefulness Novelty Coverage Reusable
'The instance of the DRA is working fine with the above technology stack.'	
'The instance of the DRA setup/ configuration is working as intended.'	
'The use and applicability of the DRA to deploy the sample demo application is working as intended. This seems to be used for other different types of IoT applications'.	Usefulness
'The DRA is working as intended for the selected hardware.'	
'Lab is bidding for **drone and robotics** application development and deployment projects. This is a huge research area and has the **potential to extend DRA**, perhaps another PhD (s), for the secure deployment of drone and robotics application projects'.	

E. Report

The case study report is an organized outcome that aims to conclude the operational proof of concept of the DRA and (LL)'s feedback about the DRA design models in the context of the DigiSAS Lab (please see Table 3). The report presents the research case and description of the case study elements. The report outlines in brief the evaluation results deduced from the analysis conducted in Table II.

TABLE III.

Case Study Analysis Report	
Research Case	***Description***
Organization	UTS SCS DigiSAS Lab
Test date	15/08/2019
Organization context	DigiSAS Lab conducts research and development in collaboration with industry partners' projects involving web and IoT and the multi-cloud.
Tester	The DigiSAS Lab leader
Organization need	DigiSAS Lab needs a multi-cloud deployment environment to meet the needs of industry partners.
Test objective	The objective is to evaluate the applicability of the DRA in the research lab environment and to test the deployment of IoT-applications to multi-cloud.
Test case	How can IoT-applications be deployed to the multi-cloud using DevOps?
Test package (Pre-prepared)	A POC demo package is prepared in the CST: Proof of concept demo YouTube video: Link Presentation slides: Link
Test component	1. DRAv2.0 architecture 2. DRA operational model pipeline 3. Software components 4. Hardware components
Test method	Case study template (please see Appendix)
Test duration	60 minutes (presentation, demo, CST)
Data type	Qualitative feedback provided by the evaluator (LL)
Pretesting	The researcher presented the case study project to the participant (LL).
Key activities	(LL) verifies that the DRA enables DevOps adoption. (LL) verifies that DRAv2.0 toolset is reusable. (LL) verifies that DRA addresses the research gaps. (LL) verifies that DRA enables automated IoT-applications deployment to multi-cloud. (LL) verifies the IoT-application-sensors interaction. (LL) verifies that DRA is instantiable and offers new knowledge-base.
Evaluation outcome	The cross-examination between the feedback (Table2) and validation criteria (Table 1) indicates that the DRA design models reusable in the research context. The evaluation indicated that DRA offers new knowledge (CI-Broker) that can be used for deploying IoT applications to the multi-cloud. Results indicate that the DRA is a general design and is not fixed to a particular situation. It can adapt to different situations and be reused with different technology stacks.

VII. Discussion And Conclusion

DevOps provide a mechanism to enable the integration of traditionally isolated development and operations capabilities in the overall context of agile [3], [15]. DevOps vendors provide a set of tools to enable the automated, fast and distributed deployment of IoT-applications to multi-cloud [1], [11]. This paper discusses the new DRA as a generic vendor independent reference architecture which enables software applications automated deployments to an integrated multi-cloud platform using the DevOps approach.

This paper presents the outcomes of an empirical evaluation of the DRA using a research case study. The evaluation results specify that the DRA framework offers a research-based functional and appropriate solution for IoT-application deployment without being influenced by vendors. The results of the evaluation indicate that the DRA framework offer sufficient consolidated and practical guidelines to practitioners and researchers and enable them to make informed decisions about the adoption of DevOps approach for IoT-applications deployment to cloud/multi-cloud. Further, the DRA evaluation highlighted several new directions for future vital research areas. This research warrants future studies in the field of DevOps for drones and robotics.

Acknowledgment

The authors wish to thank the Australian Government Research Training Program (RTP) and the Australian Government Commonwealth support and HECS-HELP for providing the funding to the research project.

References

[1] Ghantous, Georges Bou, and Asif Qumer Gill. “DevOps Reference Architecture for Multi-Cloud IoT Applications.” *2018 IEEE 20th Conference on Business Informatics (CBI)*, 2018, DOI:10.1109/cbi.2018.00026.

[2] Bou Ghantous, G., and Asif Gill. "DevOps: Concepts, practices, tools, benefits and challenges." *PACIS2017* (2017).

[3] Ghantous, Georges Bou, and Asif Qumer Gill. “An Agile-DevOps Reference Architecture for Teaching Enterprise Agile.” *International Journal of Learning, Teaching and Educational Research*, vol. 18, no. 7, 2019, pp. 128–144., DOI:10.26803/ijlter.18.7.9.

[4] Prat, Nicolas, Isabelle Comyn-Wattiau, and Jacky Akoka. "Artifact Evaluation in Information Systems Design-Science Research-a Holistic View." *PACIS*. 2014.

[5] Peffers, Ken, et al. "A design science research methodology for information systems research." *Journal of management information systems* 24.3 (2007): 45-77.

[6] Runeson, Per, and Martin Höst. “Guidelines for Conducting and Reporting Case Study Research in Software Engineering.” *Empirical Software Engineering*, vol. 14, no. 2, 2008, pp. 131–164., DOI:10.1007/s10664-008-9102-8.

[7] Carvalho, J.Á., 2012. Validation criteria for the outcomes of design research. In Pre-ECIS workshop on IT Artefact Design & Work practice Intervention.

[8] Jedlitschka, Andreas, Marcus Ciolkowski, and Dietmar Pfahl. "Reporting experiments in software engineering." *Guide to advanced empirical software engineering*. Springer, London, 2008. 201-228.

[9] Moore, John, et al. “DevOps for the Urban IoT.” *Proceedings of the Second International Conference on IoT in Urban Space - Urb-IoT '16*, 2016, DOI:10.1145/2962735.2962747.

[10] Gușeilă, Ligia Georgeta, Dragoș-Vasile Bratu, and Sorin-Aurel Moraru. "DevOps Transformation for Multi-Cloud IoT Applications." *2019 International Conference on Sensing and Instrumentation in IoT Era (ISSI)*. IEEE, 2019.

[11] Rajkumar, M, et al. “DevOps Culture and Its Impact on Cloud Delivery and Software Development.” *2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA) (Spring)*, 2016, DOI:10.1109/icacca.2016.7578902.

[12] Alonso, Juncal, et al. “DECIDE: DevOps for Trusted, Portable and Interoperable Multi-Cloud Applications towards the Digital Single Market.” *Proceedings of the 7th International Conference on Cloud Computing and Services Science*, 2017, DOI:10.5220/0006292403970404.

[13] Ferry, Nicolas, et al. “CloudMF.” *ACM Transactions on Internet Technology*, vol. 18, no. 2, 2018, pp. 1–24., DOI:10.1145/3125621.

[14] Wettinger, Johannes, et al. "Middleware-oriented deployment automation for cloud applications." *IEEE Transactions on Cloud Computing* 6.4 (2016): 1054-1066.

[15] Bai, Xiaoying, et al. "The DevOps Lab Platform for Managing Diversified Projects in Educating Agile Software Engineering." *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE, 2018.

[16] Luz, Welder Pinheiro, et al. “Adopting DevOps in the Real World: A Theory, a Model, and a Case Study.” *Journal of Systems and Software*, vol. 157, 2019, p. 110384., DOI:10.1016/j.jss.2019.07.083.

[17] Kurdi, Heba, et al. “MultiCuckoo: Multi-Cloud Service Composition Using a Cuckoo-Inspired Algorithm for the Internet of Things Applications.” *IEEE Access*, vol. 6, 2018, pp. 56737–56749., DOI:10.1109/access.2018.2872744.

[18] Ferry, Nicolas, and Phu H. Nguyen. “Towards Model-Based Continuous Deployment of Secure IoT Systems.” *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, 2019, DOI:10.1109/models-c.2019.00093.

[19] Wettinger, Johannes, et al. “Streamlining DevOps Automation for Cloud Applications Using TOSCA as Standardized Metamodel.” *Future Generation Computer Systems*, vol. 56, 2016, pp. 317–332., DOI:10.1016/j.future.2015.07.017.

[20] Chang, Carl K. “Agile, Continuous Integration, and DevOps.” *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, 2019, DOI:10.1109/compsac.2019.00038.

[21] Hamza, Muhammad, et al. “Siot-Rimm.” *Proceedings of the Evaluation and Assessment in Software Engineering*, 2020, DOI:10.1145/3383219.3383286.

[22] Sanjeevi, P., et al. “A Performance-Aware Dynamic Scheduling Algorithm for Cloud-Based IoT Applications.” *Computer Communications*, 2020, DOI:10.1016/j.comcom.2020.06.016.

[23] Douzis, Konstantinos, et al. “Modular and Generic IoT Management on the Cloud.” *Future Generation Computer Systems*, vol. 78, 2018, pp. 369–378., DOI:10.1016/j.future.2016.05.041.

[24] Yonezawa, Takuro, et al. “SOXFire.” *Proceedings of the 2nd International Workshop on Smart - SmartCities '16*, 2016, DOI:10.1145/3009912.3009922.

[25] Naveen, Soumyalatha, and Manjunath R Kounte. “Key Technologies and Challenges in IoT Edge Computing.” *2019 Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2019, DOI:10.1109/i-smac47947.2019.9032541.

[26] Shi, Tao, et al. “A Genetic-Based Approach to Location-Aware Cloud Service Brokering in Multi-Cloud Environment.” *2019 IEEE International Conference on Services Computing (SCC)*, 2019, DOI:10.1109/scc.2019.00034.

[27] Jula, Amin, et al. “Cloud Computing Service Composition: A Systematic Literature Review.” *Expert Systems with Applications*, vol. 41, no. 8, 2014, pp. 3809–3824., DOI:10.1016/j.eswa.2013.12.017.

[28] Kritikos, Kyriakos, and Dimitris Plexousakis. “Multi-Cloud Application Design through Cloud Service Composition.” *2015 IEEE 8th International Conference on Cloud Computing*, 2015, DOI:10.1109/cloud.2015.96.

[29] Yasrab, Robail, and Naijie Gu. “Multi-Cloud PaaS Architecture (MCPA): A Solution to Cloud Lock-In.” *2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*, 2016, DOI:10.1109/icisce.2016.108.

[30] Slawik, Mathias, et al. “CYCLONE: The Multi-Cloud Middleware Stack for Application Deployment and Management.” *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2017, DOI:10.1109/cloudcom.2017.56.

[31] Jamshidi, Pooyan, et al. “Cloud Migration Patterns: A Multi-Cloud Service Architecture Perspective.” *Service-Oriented Computing - ICSOC 2014 Workshops Lecture Notes in Computer Science*, 2015, pp. 6–19., DOI:10.1007/978-3-319-22885-3_2.

[32] Martino, Beniamino Di, and Antonio Esposito. “Semantic Techniques for Multi-Cloud Applications Portability and Interoperability.” *Procedia Computer Science*, vol. 97, 2016, pp. 104–113., DOI:10.1016/j.procs.2016.08.285.

[33] Reshi, Aijaz Ahmad, et al. “Development and Web Performance Evaluation of Internet of Things Testbed.” *2019 International Conference on Computer and Information Sciences (ICCIS)*, 2019, DOI:10.1109/iccisci.2019.8716436.

[34] Vyas, Sonali, and Deepshikha Bhargava. “Big Data Utilization, Benefits, and Challenges for Smart City Implementation.” Advances in Data Mining and Database Management Handbook of Research on Big Data and the IoT, 2019, pp. 42–54., DOI:10.4018/978-1-5225-7432-3.ch003.

Appendices

1. Digital Strategy, Architecture & Solutions (SCS DigiSAS Lab): http://www.digisaslab.org/
2. CloudStor from AARNet (recommended UTS cloud storage): https://www.aarnet.edu.au/
3. DevOps Reference Architecture (DRA) project page: https://maven-app-heroku.herokuapp.com/
4. Case Study Template (CST): Link https://cloudstor.aarnet.edu.au/plus/s/LQvcR2P1YmoDs4B
5. DigiSAS Research Lab evaluation data: Link https://cloudstor.aarnet.edu.au/plus/s/BdkVGQaSzqGLEnm