Testing Patterns that Correlate With Appearance of Bugs - Similarities and Differences: From Section IV-B1 we identify three testing patterns that correlate with appearance of bugs. Of these three patterns assertion roulette and remote mystery guest have been documented as test smells in existing research [8], [66]. The implication of this finding is that the testing patterns that have negative implications for GPLs can also occur for non-GPLs, such as for Ansible test scripts.

We also notice one testing pattern to correlate with appearance of bugs that is unique to Ansible: local only testing. An IaC test script may execute correctly in a local environment, but erroneously in a remote environment, potentially leading to erroneous provisioning of computing infrastructure [27]. Practitioners consider the activity of conducting IaC testing on remote environments as a good practice stating "*by running the tests on real systems, you can determine whether your application responded correctly in a realistic configuration*" [61].

Implications for Reproducible Deployments: One of the perceived benefits of IaC is reproducible deployments of cloud-based infrastructure, which enables practitioners to provision cloud-based infrastructure with consistent environments [33]. However, as shown in Section IV-B, practitioners use local only testing, which can undermine the value of IaC with respect to reproducible deployments. The example presented in Listing 4 tests functionality of network bridges only in the local development environment. The network bridge functionality may behave correctly for the local environment but not for one or multiple remote cloud instances due to differences in system configurations, package dependencies, etc., potentially creating inconsistencies between local and cloud-based environments. Local only testing is symptomatic of an 'uncontrollable configuration management process' [33], and is considered as a deterrent for reproducible deployments [10].

Implications for Troubleshooting Test Failures: From Section IV-B we observe test scripts can have as many as 25 assertions under a single `assert` tag. Existence of assertion roulette instances can negatively impact comprehension of test failures [8], which can make troubleshooting of test failures harder. Troubleshooting failures in cloud-based software development is challenging [19], and instances of assertion roulette can further aggravate the challenges that are related with cloud-based infrastructure maintenance.

Future Directions: Researchers can develop techniques that will investigate run-time behavior of test scripts and characterize potential flakiness in test scripts. Researchers can also investigate if other categories of testing patterns, which correlate with appearance of bugs, exist for Ansible scripts as well as for Chef, Puppet, and Terraform scripts.

VI. Threats to Validity

We discuss the limitations of our paper as follows:

Conclusion Validity: Our identified bug categories and testing patterns are limited to the commits within the dataset we used in Section III-A1. Also, the set of 500 Ansible test scripts used in Section III-B1 to determine testing patterns is subject to the first author's bias. The identified bug and testing pattern categories are susceptible to rater bias, which we mitigate by using two raters. We use commits to identify bug categories and bug-related test scripts, which can be limiting. Also, TAMA can generate false negatives and false positives when applied on other datasets. We mitigate this limitation by evaluating TAMA using an oracle dataset described in Section III-B2. Furthermore, results presented in Section IV-B show a correlation between testing patterns and appearance of bugs, but such correlation will not always lead to causation.

External Validity: Our datasets are constructed by mining OSS repositories. Our findings may not generalize for proprietary datasets. Also, our findings are limited to IaC scripts developed using Ansible, which may not generalize to other IaC languages, such as Chef and Puppet.

Internal Validity: While constructing the oracle dataset the rater may have expectations on the outcomes that could potentially impact the closed coding process. We mitigate the limitation by using a rater who is not an author of the paper. Furthermore, construction of the oracle dataset is susceptible to raters' experience in Ansible. We mitigate this limitation by providing the rater a document that describes each pattern with definitions and examples.

VII. Conclusion

The practice of IaC advocates for integrating quality into IaC development and testing. A characterization study of bugs in IaC test scripts, such as Ansible test scripts, is the first step towards aiding practitioners on how to integrate quality into IaC testing. Such characterization can also identify testing patterns that correlate with appearance of bugs in test scripts. We have conducted an empirical study with 4,831 Ansible test scripts mined from 104 OSS repositories. We observe bugs to appear in 1.8% of the 4,831 Ansible test scripts in our dataset. We identify 7 bug categories: configuration, dependency, idempotency, logging, performance, security, and style. We also identify 3 testing patterns that correlate with appearance of bugs of which local only testing is unique to Ansible test scripts.

Based on our findings, we recommend application of techniques and tools that target one or more of our identified bug categories. We also recommend the use of TAMA to identify instances of the 3 testing patterns, as detection of testing patterns can help practitioners prioritize inspection efforts to find bugs in Ansible test scripts. We hope our paper will facilitate more research in the domain of IaC script quality.

Acknowledgments

We thank the PASER group at Tennessee Tech University for their valuable feedback. The research was partially funded by the U.S. National Science Foundation (NSF) award # 2026869.

References

[1] A. Agrawal, A. Rahman, R. Krishna, A. Sobran, and T. Menzies, "We don't need another hero?: The impact of "heroes" on software development," in *Proceedings of the 40th International Conference on Software Engineering: Software Engineering in Practice*, ser. ICSE-SEIP '18. New York, NY, USA: ACM, 2018, pp. 245–253. [Online]. Available: http://doi.acm.org/10.1145/3183519.3183549

[2] akondrahman, "akondrahman/IaCTesting," https://github.com/akondrahman/IaCTesting, 2021, [Online; accessed 26-Dec-2021].

[3] Alison DeNisco Rayome, "Ansible overtakes Chef and Puppet as the top cloud configuration management tool," https://www.techrepublic.com/article/ansible-overtakes-chef-and-puppet-as-the-top-cloud-configuration-management-tool/, 2019, [Online; accessed 25-Sep-2021].

[4] Ansible, "Ansible Documentation," https://docs.ansible.com/, 2021, [Online; accessed 19-Sep-2021].

[5] ——, "Ansible Lint Documentation," https://ansible-lint.readthedocs.io/en/latest/, 2021, [Online; accessed 29-Sep-2021].

[6] ——, "Swisscom Automates IT Management WITH RedHat Ansible Tower," https://www.ansible.com/hubfs/pdfs/RH-Ansible-Tower-swisscom-case-study.pdf?hsLang=en-us, 2021, [Online; accessed 13-Sep-2021].

[7] T. Barik, D. Ford, E. Murphy-Hill, and C. Parnin, "How should compilers explain problems to developers?" in *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 2018, pp. 633–643.

[8] G. Bavota, A. Qusef, R. Oliveto, A. De Lucia, and D. Binkley, "An empirical analysis of the distribution of unit test smells and their impact on software maintenance," in *2012 28th IEEE International Conference on Software Maintenance (ICSM)*, 2012, pp. 56–65.

[9] B. Beizer, *Software system testing and quality assurance*. Van Nostrand Reinhold Co., 1984.

[10] Y. Brikman, "5 lessons learned from writing over 300,000 lines of infrastructure code," https://blog.gruntwork.io/5-lessons-learned-from-writing-over-300-000-lines-of-infrastructure-code-36ba7fadeac1, 2018, [Online; accessed 14-Sep-2021].

[11] CentOS-PaaS-SIG/linchpin, "Update unit tests contra-hdsl," https://github.com/CentOS-PaaS-SIG/linchpin/commit/4905430ab36c, 2019, [Online; accessed 25-August-2021].

[12] ceph/ceph ansible, "tests: resize root partition when atomic host," https://github.com/ceph/ceph-ansible/commit/e1c1017e15, 2018, [Online; accessed 21-Jun-2021].

[13] ——, "ceph-ansible:Ansible playbooks to deploy Ceph, the distributed filesystem." https://github.com/ceph/ceph-ansible, 2021, [Online; accessed 23-Jun-2021].

[14] ——, "Use ansible_facts," https://github.com/ceph/ceph-ansible/commit/7ddbe747122, 2021, [Online; accessed 24-Jun-2021].

[15] ceph/ceph installer, "tests: remove duplicate logging statements," https://github.com/ceph/ceph-installer/commit/634cdc8b1f, 2017, [Online; accessed 24-Jun-2021].

[16] R. Chillarege, I. Bhandari, J. Chaar, M. Halliday, D. Moebus, B. Ray, and M.-Y. Wong, "Orthogonal defect classification-a concept for in-process measurements," *IEEE Transactions on Software Engineering*, vol. 18, no. 11, pp. 943–956, Nov 1992.

[17] Chris Meyers, "Five Questions: Testing Ansible Playbooks & Roles," https://www.ansible.com/blog/five-questions-testing-ansible-playbooks-roles, 2017, [Online; accessed 22-Sep-2021].

[18] M. Cinque, D. Cotroneo, R. D. Corte, and A. Pecchia, "Assessing direct monitoring techniques to analyze failures of critical industrial systems," in *2014 IEEE 25th International Symposium on Software Reliability Engineering*, Nov 2014, pp. 212–222.

[19] J. Cito, P. Leitner, T. Fritz, and H. C. Gall, "The making of cloud applications: An empirical study on software development for the cloud," in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, ser. ESEC/FSE 2015. New York, NY, USA: Association for Computing Machinery, 2015, p. 393–403. [Online]. Available: https://doi.org/10.1145/2786805.2786826

[20] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, vol. 20, no. 1, pp. 37–46, 1960. [Online]. Available: http://dx.doi.org/10.1177/001316446002000104

[21] D. Cramer and D. L. Howitt, *The Sage dictionary of statistics: a practical resource for students in the social sciences*. Sage, 2004.

[22] J. Davila, "Ansible/NASA Case Study," http://szsb-gl2x.accessdomain.com/fierce/wp-content/uploads/2016/01/NASA-Case-Study-Ansible.pdf, 2016, [Online; accessed 20-Jun-2021].

[23] G. Dhillon and J. Backhouse, "Current directions in is security research: towards socio-organizational perspectives," *Information systems journal*, vol. 11, no. 2, pp. 127–153, 2001.

[24] Docker, "Registry as a pull through cache," https://docs.docker.com/registry/recipes/mirror/, 2021, [Online; accessed 25-August-2021].

[25] DockerHub, "Build and Ship any Application Anywhere," https://hub.docker.com/, 2021, [Online; accessed 26-August-2021].

[26] M. Guerriero, M. Garriga, D. A. Tamburri, and F. Palomba, "Adoption, support, and challenges of infrastructure-as-code: Insights from industry," in *2019 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2019, pp. 580–589.

[27] M. M. Hasan, F. A. Bhuiyan, and A. Rahman, "Testing practices for infrastructure as code," in *Proceedings of the 1st ACM SIGSOFT International Workshop on Languages and Tools for Next-Generation Testing*, 2020, pp. 7–12.

[28] M. M. Hassan and A. Rahman, "Verifiability package for paper," https://figshare.com/s/4aa50ec7c34c18c71223, 2021, [Online; accessed 01-Jan-2022].

[29] M. M. Hennink, B. N. Kaiser, and V. C. Marconi, "Code saturation versus meaning saturation: how many interviews are enough?" *Qualitative health research*, vol. 27, no. 4, pp. 591–608, 2017.

[30] R. Hersher, "Incident documentation/20170118-Labs," https://www.npr.org/sections/thetwo-way/2017/03/03/518322734/amazon-and-the-150-million-typo, 2017, [Online; accessed 21-Sep-2021].

[31] F. Hoffa, "GitHub on BigQuery: Analyze all the open source code," https://cloud.google.com/blog/products/gcp/github-on-bigquery-analyze-all-the-open-source-code, 2016, [Online; accessed 16-Dec-2020].

[32] N. Humbatova, G. Jahangirova, G. Bavota, V. Riccio, A. Stocco, and P. Tonella, "Taxonomy of real faults in deep learning systems," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, ser. ICSE '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1110–1121. [Online]. Available: https://doi.org/10.1145/3377811.3380395

[33] J. Humble and D. Farley, *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation*, 1st ed. Addison-Wesley Professional, 2010.

[34] W. Hummer, F. Rosenberg, F. Oliveira, and T. Eilam, "Automated testing of chef automation scripts," in *Proceedings Demo & Poster Track of ACM/IFIP/USENIX International Middleware Conference*, 2013, pp. 1–2.

[35] ——, "Testing idempotence for infrastructure as code," in *Middleware 2013*, D. Eyers and K. Schwan, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 368–388.

[36] IEEE, "IEEE standard classification for software anomalies," *IEEE Std 1044-2009 (Revision of IEEE Std 1044-1993)*, pp. 1–23, Jan 2010.

[37] Y. Jiang and B. Adams, "Co-evolution of infrastructure and source code: An empirical study," in *Proceedings of the 12th Working Conference on Mining Software Repositories*, ser. MSR '15. Piscataway, NJ, USA: IEEE Press, 2015, pp. 45–55. [Online]. Available: http://dl.acm.org/citation.cfm?id=2820518.2820527

[38] kubernetes sigs/kubespray, "Security best practice fixes," https://github.com/kubernetes-sigs/kubespray/commit/d487b2f9279, 2017, [Online; accessed 24-Jun-2021].

[39] ——, "Refactor download role," https://github.com/kubernetes-sigs/kubespray/commit/66408a87ee, 2020, [Online; accessed 24-Jun-2021].

[40] P. Labs, "Puppet Documentation," https://docs.puppet.com/, 2021, [Online; accessed 08-Aug-2021].

[41] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977. [Online]. Available: http://www.jstor.org/stable/2529310

[42] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The Annals of Mathematical Statistics*, vol. 18, no. 1, pp. 50–60, 1947. [Online]. Available: http://www.jstor.org/stable/2236101

[43] M. N. Marshall, "Sampling for qualitative research," *Family practice*, vol. 13, no. 6, pp. 522–526, 1996.

[44] K. Morris, *Infrastructure as code: managing servers in the cloud*. " O'Reilly Media, Inc.", 2016.

[45] N. Munaiah, S. Kroh, C. Cabrey, and M. Nagappan, "Curating github for engineered software projects," *Empirical Software Engineering*, pp. 1–35, 2017. [Online]. Available: http://dx.doi.org/10.1007/s10664-017-9512-6

[46] openstack/openstack ansible, "Fix idempotency bug in AIO bootstrap," https://github.com/openstack/openstack-ansible/commit/a4dfb65169, 2016, [Online; accessed 24-Jun-2021].

[47] ——, "Use operating system specific IP utilities," https://github.com/openstack/openstack-ansible/commit/b697c55842, 2018, [Online; accessed 20-Jun-2021].

[48] openstack/openstack-ansible lxc_hosts, "Fix ansible-lint errors," https://github.com/openstack/openstack-ansible-lxc_hosts/commit/0d28eeab560, 2021, [Online; accessed 26-August-2021].

[49] os-cloud/openstack-ansible galera_server, "Updated repo for new org," https://github.com/os-cloud/openstack-ansible-galera_server/commit/cd11c5a56e96c0, 2015, [Online; accessed 27-August-2021].

[50] os-cloud/os-ansible deployment, "Fix main public interface name not always be eth0." https://github.com/os-cloud/os-ansible-deployment/commit/03d176d5a, 2016, [Online; accessed 22-Jun-2021].

[51] L. A. Palinkas, S. M. Horwitz, C. A. Green, J. P. Wisdom, N. Duan, and K. Hoagwood, "Purposeful sampling for qualitative data collection and analysis in mixed method implementation research," *Administration and policy in mental health and mental health services research*, vol. 42, no. 5, pp. 533–544, 2015.

[52] A. Peruma, K. Almalki, C. D. Newman, M. W. Mkaouer, A. Ouni, and F. Palomba, "Tsdetect: An open source test smells detection tool," in *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. ESEC/FSE 2020. New York, NY, USA: Association for Computing Machinery, 2020, p. 1650–1654. [Online]. Available: https://doi.org/10.1145/3368089.3417921

[53] A. Rahman, F. L. Barsha, and P. Morrison, "Shhh!: 12 practices for secret management in infrastructure as code," in *2021 IEEE Secure Development Conference (SecDev)*, 2021, pp. 56–62.

[54] A. Rahman, E. Farhana, C. Parnin, and L. Williams, "Gang of eight: A defect taxonomy for infrastructure as code scripts," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, ser. ICSE '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 752–764. [Online]. Available: https://doi.org/10.1145/3377811.3380409

[55] A. Rahman, E. Farhana, and L. Williams, "The 'as code'activities: development anti-patterns for infrastructure as code," *Empirical Software Engineering*, vol. 25, no. 5, pp. 3430–3467, 2020.

[56] A. Rahman, C. Parnin, and L. Williams, "The seven sins: security smells in infrastructure as code scripts," in *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*. IEEE, 2019, pp. 164–175.

[57] A. Rahman, M. R. Rahman, C. Parnin, and L. Williams, "Security smells in ansible and chef scripts: A replication study," *ACM Trans. Softw. Eng. Methodol.*, vol. 30, no. 1, Jan. 2021. [Online]. Available: https://doi.org/10.1145/3408897

[58] A. A. U. Rahman and E. Farhana, "An empirical study of bugs in covid-19 software projects," *Journal of Software Engineering Research and Development*, vol. 9, no. 1, p. 3:1 – 3:19, Mar. 2021. [Online]. Available: https://sol.sbc.org.br/journals/index.php/jserd/article/view/827

[59] L. Ryzhyk, P. Chubb, I. Kuz, and G. Heiser, "Dingo: Taming device drivers," in *Proceedings of the 4th ACM European conference on Computer systems*, 2009, pp. 275–288.

[60] J. Saldaña, *The coding manual for qualitative researchers*. Sage, 2015.

[61] D. Schmitt, "Hitchhiker's guide to testing infrastructure as/and code — don't panic!" https://puppet.com/blog/hitchhikers-guide-to-testing-infrastructure-as-and-code/, 2016, [Online; accessed 20-Jun-2021].

[62] J. Schwarz, A. Steffens, and H. Lichter, "Code smells in infrastructure as code," in *2018 11th International Conference on the Quality of Information and Communications Technology (QUATIC)*, 2018, pp. 220–228.

[63] C. B. Seaman, F. Shull, M. Regardie, D. Elbert, R. L. Feldmann, Y. Guo, and S. Godfrey, "Defect categorization: Making use of a decade of widely varying historical data," in *Proceedings of the Second ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 149–157. [Online]. Available: https://doi.org/10.1145/1414004.1414030

[64] T. Sharma, M. Fragkoulis, and D. Spinellis, "Does your configuration code smell?" in *Proceedings of the 13th International Conference on Mining Software Repositories*, ser. MSR '16. New York, NY, USA: ACM, 2016, pp. 189–200. [Online]. Available: http://doi.acm.org/10.1145/2901739.2901761

[65] talismanic, "akondrahman/IaCTesting," https://hub.docker.com/r/talismanic/tama, 2021, [Online; accessed 26-Jan-2022].

[66] M. Tufano, F. Palomba, G. Bavota, M. Di Penta, R. Oliveto, A. De Lucia, and D. Poshyvanyk, "An empirical investigation into the nature of test smells," in *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE 2016. New York, NY, USA: Association for Computing Machinery, 2016, p. 4–15. [Online]. Available: https://doi.org/10.1145/2970276.2970340

[67] A. Van Deursen, L. Moonen, A. Van Den Bergh, and G. Kok, "Refactoring test code," in *Proceedings of the 2nd international conference on extreme programming and flexible processes in software engineering (XP)*, 2001, pp. 92–95.

[68] A. Weiss, A. Guha, and Y. Brun, "Tortoise: Interactive system configuration repair," in *Proceedings of the 32Nd IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE 2017. Piscataway, NJ, USA: IEEE Press, 2017, pp. 625–636. [Online]. Available: http://dl.acm.org/citation.cfm?id=3155562.3155641