DEFECTS-FREE INFRASTRUCTURE AS CODE IMPLEMENTATION WITH ANSIBLE BASED ON CODE METRICS ANALYSIS

Simkin Andrii, Kopp Andrii
National Technical University «Kharkiv Polytechnic Institute», Kharkiv

We live in an age of automation, when anyone may start an entire application server with a single click. It will contain the required operating system, programs, security restrictions, hard-drive space, and a plethora of unique options. When the team need it, the server setup may be duplicated dozens, hundreds, or even thousands of times without human involvement. All of this is achievable using the Infrastructure as Code (IaC) methodology [1].

Instead of manually configuring each server, DevOps or system engineers construct a configuration script with all the necessary settings and modifications, and a provisioning tool does the rest.

An example of using Ansible [2], one of the most popular software provisioning tools, through the GitHub repository is demonstrated below (Fig. 1).

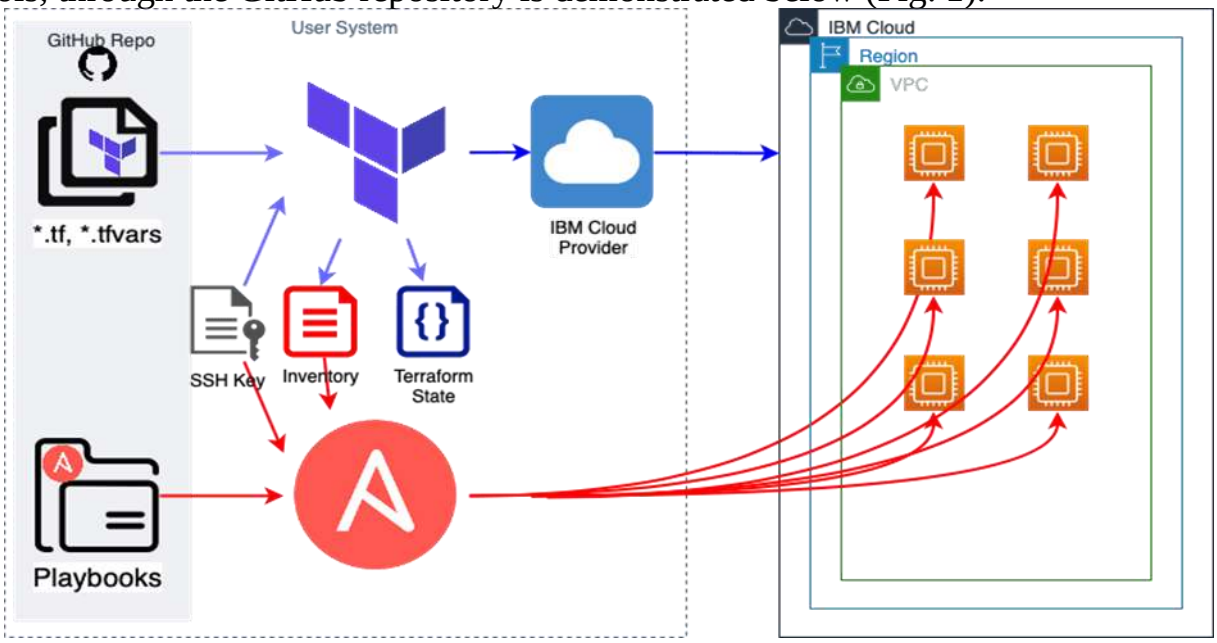

Fig. 1. – Ansible playbook deployment process [2]

However, IaC scripts created for Ansible or any other provisioning tools still include human-written source code. Therefore, IaC scripts can contain defects that negatively impact software infrastructure causing time and cost expenses. One of the possible ways to detect defective IaC scripts and prevent them from execution is code metrics analysis, such as lines of code (LOC) and others [1].

References:

1. Poster: Defect Prediction Metrics for Infrastructure as Code Scripts in DevOps // URL: https://akondrahman.github.io/files/papers/icse18_poster.pdf
2. Ansible // URL: https://ibm.github.io/cloud-enterprise-examples/iac-conf-mgmt/ansible/