A Cloud Based Lab Environment for Network Security Research

Franklin E. Diaz
fdiaz@paloaltonetworks.com[1,2]

[1]Palo Alto Networks
[2]Professional Services - Automation

September 18, 2021

Abstract

The goal of this document is to detail the major components of the "Cloudlab" network security lab in GCP. The reasoning behind including certain components and design elements is explained. The relationship between security and the lab are detailed to highlight purpose. Security learning objectives are realized at every step of the process in the creation and use of this lab. The result is a widely scoped and highly flexible security learning environment for data scientists and engineers with focus on development, test, operations, and continuous pipelines.

1 A Cloud Native Security Lab

There is a proliferation of new and constantly evolving Public Cloud providers, tools, tool chains, and whole "cloud based" development and operations ecosystems. There is a need to understand and adapt to the "GitOps" paradigm and the business opportunities this fundamental shift presents. There is no disputing the fact that this shift is well underway[4].

> *Cloud-native technologies empower organizations to build and run scalable applications in modern, dynamic environments such as public, private, and hybrid clouds.*
>
> Cloud Native Computing Foundation

The "Cloudlab" is a private lab environment for security research, testing, and training. The lab is "Cloud Native" in the sense that it adheres to GitOps practices. The GitOps methodology facilitates storage, review, and maintenance of the lab Infrastructure as Code. The lab is currently comprised of two main Pipelines. The first of these is dedicated to Continuous Integration. The second pipeline facilitates Continuous Machine Learning, or CML[1].

As security practitioners, it is important to understand an environment as a precursor to securing it.

2 Project Goals

There are several goals related to this project. These goals overlap and interconnect at times. The overarching goal is to understand the composition, management, and weaknesses of the items listed here.

1. Provisioning Palo Alto Networks CN-Series firewall products, integrating them with Calico and protecting containerized workloads (Kubernetes "pods").
2. Research containerized deployments, workloads and security.
3. Demonstrate "automation bots" and how they interact with a GitHub repository. Extending a revision control platform (GitHub in this case) is rather common in large organizations.
4. Integration of Bridgecrew "Security as Code" tooling with GitHub repositories.
5. Develop and demonstrate "serverless cloud function" expertise (GCP Cloud Functions/AWS Lambda)
6. Develop and use a cloud native Continuous Integration build pipeline. The output of this pipeline is a Docker image that is stored in gcr.io. These images include a fully contained set of tools, documentation and Terraform code for customer deployments.

Download latest version

7. Demonstrate Policy as Code concepts using Terratest and Kyverno.

3 Lab Configuration Details

The configuration files and code used to build the lab are stored on GitHub. The lab is running on Google Cloud, one of the "big three" public cloud providers. The lab is intended to server as a proof of concept as well as provide a teaching and training platform.

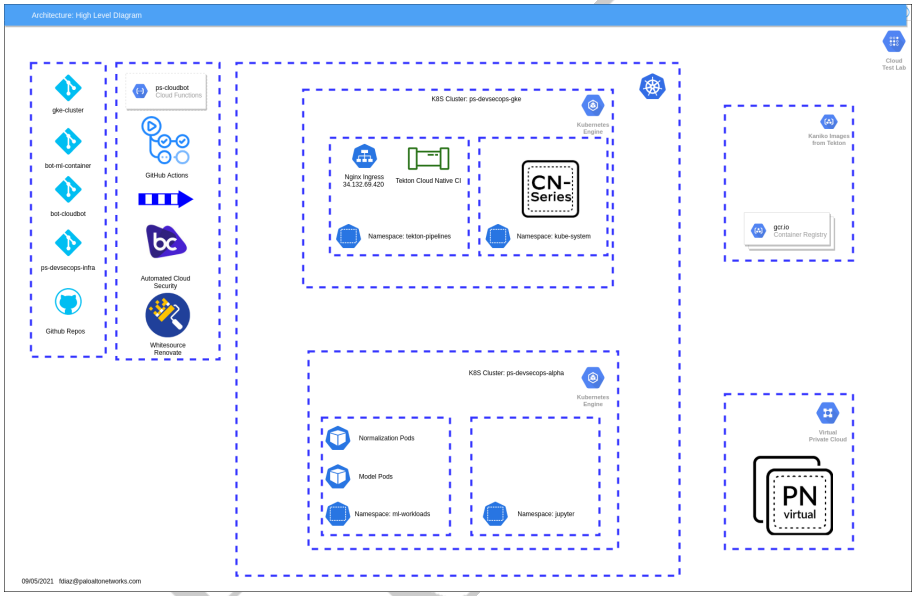

Figure 1: High Level Lab Design Diagram

A detailed description of the five main blocks in figure 1 follows.

1 **[GitHub Repositories]**
The code base for the lab is broken down into several GitHub repositories, more or less around the functional area.

Repo Name	Purpose
bot-cloudbot	A custom Python 3.9 GCP Cloud Function for GitHub pull request task automation.
bot-ml-container	An experimental containerized machine learning model.
gke-cluster	Infra as Code files for GKE cluster, YAML files for Tekton CI pipeline. CN Series firewall nodes.
ps-containerizer	An 'invisible shim" with a Docker image for each "public cloud" VM-Series Terraform module development repo. Allows PRs to be ingested into Tekton CI pipeline without any integrations with the source repo.
ps-devsecops-alpha	IaC files for Alpha K8s cluster.
python-project-template	Python project template for writing serverless code in AWS Lambda and GCP cloud functions.

Table 1: Project Codebase - GitHub Repositories

2 **[GitHub Actions]** GitHub repositories that have been "on-boarded" to the project have certain "actions" included.

1. Bridgecrew is used to scan all commits to all open pull requests.
2. Whitesource renovate is used to track keep project dependencies up to date and secure.
3. Another GitHub action defines the parameters of the GCP project and helps the "ps-cloudbot" with pull request maintenance tasks.

Note that there is also a webhook to make the CI pipeline aware of each commit and kick off a test and build cycle.

3 **[Kubernetes Clusters]** Two clusters are deployed with the Google Kubernetes Engine (GKE). The "gke" cluster hosts the Tekton CI pipeline. The "alpha" cluster is used for machine learning experimentation.

Pipeline runs can be viewed and managed through a graphical interface that is well suited for development teams. There is also the ability to manage pipeline runs and their requisite tasks using standard command line tooling.

3.1: Example pipeline command

```
franklin ~/gke: tkn pr ls
NAME                STARTED          DURATION     STATUS
gh-pr-run-ncwrr     1 hour ago       1 minute     Succeeded
gh-pr-run-vhrvs     2 hours ago      1 minute     Succeeded
gh-pr-run-q6szq     3 hours ago      1 minute     Succeeded
gh-pr-run-8vn22     6 hours ago      1 minute     Succeeded
gh-pr-run-h7twc     14 hours ago     17 seconds   Failed
gh-pr-run-74wm7     21 hours ago     1 minute     Succeeded
gh-pr-run-tqlfg     1 day ago        1 minute     Succeeded
gh-pr-run-xs52f     1 day ago        1 minute     Succeeded
gh-pr-run-xtdz6     1 day ago        1 minute     Succeeded
gh-pr-run-w2zn7     1 day ago        1 minute     Succeeded
```

4 **[GCR Private Container Repository]** To date there are about 15 GitHub repositories that are integrated with the CI pipeline outlined in this paper. Each commit to a pull request causes the CI pipeline to generate a Docker image. These Docker images are stored in a private GCR location.

5 **[Panorama Management]** Panorama is a key component in deployment and maintenance of Kubernetes clusters and CN Series firewalls. Currently we have two virtual Panorama devices deployed in a high availability (HA) configuration.

Palo Alto networks has historically maintained AWS Lambda functions for firewall "auto-scaling" tasks. This has since been replaced by a set of official "plug-ins" that can be downloaded to the Panorama and NGFW devices. Lambda and Cloud functions are still frequently used to augment the capabilities of this new family of plug-ins.

4 Learning Objectives

Building and operating this lab is meant to foster learning. It may be beneficial for engineers to focus on some or all of the following list of forward-looking topics.

1. Deployment and operation of CN Series firewalls.
2. Kubernetes role based access control (RBAC) and security.
3. Developing serverless functions in Python.
4. Containers and container security.

Training materials derived from lab construction and operation are easily within reach of PS engineers.

5 Continuous Integration

Keeping internal and external build pipelines secure, as well as scanning work products that traverse these pipelines is highly desirable. To that end, a fully operational Cloud Native Continuous Integration pipeline has been implemented in the lab.

The pipeline can ingest a code base from a public repo, or a private repo with proper credentials. The "containerizer" repo demonstrates the ability to collect files from a repository, bundle it with requisite command line tools, test cases, and other necessities, and ship the image to the gcr.io container registry at the conclusion of a successful pipeline run. a pipeline run refers to a set of test cases managed and executed by Tekton.

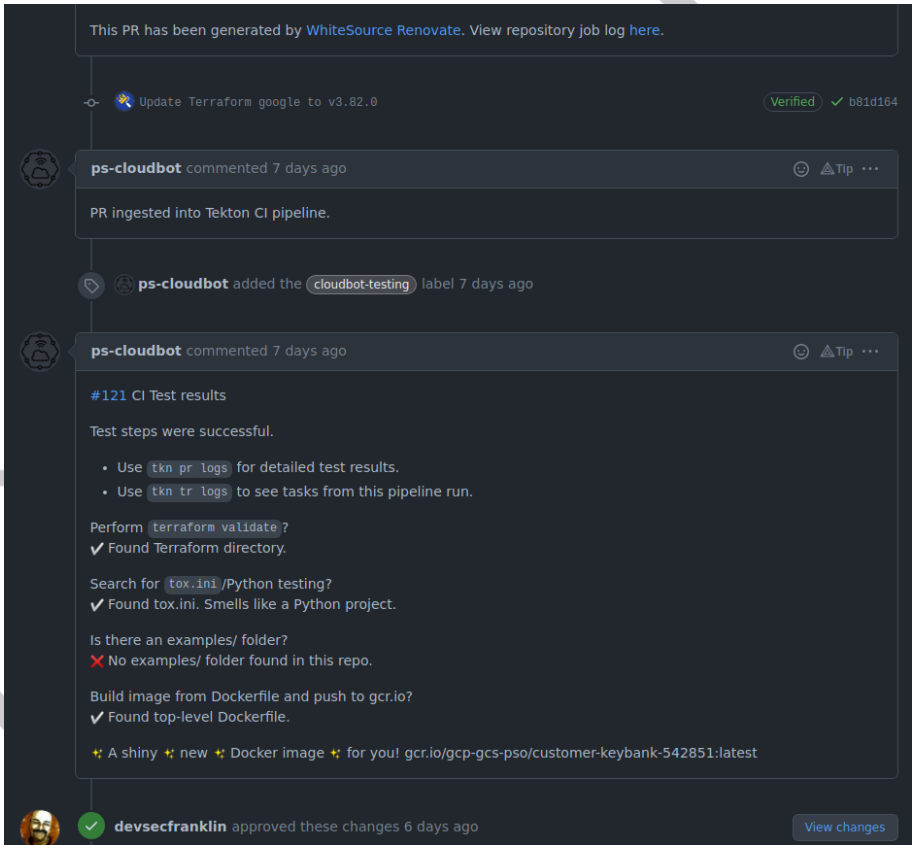

Figure 2: Tekton automated pipeline run results in a GitHub comment

Consider figure 2. The "renovate" bot detects an out of date or insecure dependency. A pull request is opened by the renovate bot. Next, "ps-cloudbot" GCP cloud function is notified of the new pull request via webhook. The second bot places a label on the pull request to indicate it is performing administrative actions on the pull request. The cloud function might perform other actions such as assigning the pull request to a certain user, adding certain users as reviewers, providing documentation, and so on. In this example, the cloud function adds a comment to the pull request to inform project members that it has been accepted by the CI pipeline. A set of test cases based on certain technology or functional area is executed and the results are returned ot the pull request in a second comment. There is also a link to the container image in the container registry. Although the pull request is merged manually in this instance, the workflow could be modified to approve and merge the pull request with no human interaction whatsoever.

Figure 3: Bridgecrew integration with pull request automation

In addition to the ability to scan for dependencies which are out of date or have vulnerabilities of note, we can perform automated security scanning of the code base as seen in figure 3. As before, the renovate bot has detected the use of an out of date Docker base image for Golang. Because the Dockerfile is updated as part of this pull request, Bridgecrew scans the file and triggers on misconfigurations that may lead to security issues. The results of the scans and the issues are noted in the pull request comments. Automated remediation and merging of these issues may be possible in some cases.

6 Summary

Future goals for the lab project might include, but are not limited to the following list.

1. Orchestration of Multi-Cloud infrastructure builds and deployments, using Crossplane for example.
2. Understanding infrastructure as code in languages besides HCL (aka Terraform). One example of this is Pulumi.
3. "Red teaming" our own lab, conducting dynamic application security testing (DAST) exercises.

References

[1] Ethem Alpaydin. *Machine Learning: The New AI*. MIT Press, 2016.

[2] Brendan Burns. *Kubernetes best practices : blueprints for building successful applications on Kubernetes*. O'Reilly Media, Sebastopol, CA, 2019.

[3] Gene Kim. *The DevOps handbook : how to create world-class agility, reliability, & security in technology organizations*. IT Revolution Press, LLC, Portland, OR, 2016.

[4] Christy Pettey. Cloud shift impacts all it markets. https://www.gartner.com/smarterwithgartner/cloud-shift-impacts-all-it-markets/, 2020. [Online; accessed 2021-09-05].

[5] Dan Sullivan. *Official Google Professional Cloud Architect : study guide*. Sybex, a Wiley brand, Hoboken, NJ, 2020.