Project Infrastructure as Code

Oscar Alexander

Contents

Project Definitie

Het doel van dit project is om een kant-en-klare cloud-ready open source oplossing te bouwen. Mijn gedacht was om een Sandbox te maken er verschillende mogelijkheden aan toevoegen. Ik wil de user de mogelijkheid geven om de Sandbox te gebruiken om verschillende tests uit te voeren op files, zoals security scans, integriteit tests, performantie tests, analyse op patronen, keywords en attributen, en misschien een machine learning model om anomalieën te detecteren.

Hieronder is een overzicht van de tests, en de opensource technologieën die ik wil gebruiken om deze tests tot werking te brengen.

- Antivirus scanning: Clamav
- Integriteit checks: Haslib
- Execution time: Time, timeit
- Performance test: Psutil
- Patroon herkenning: Regular expressies (re)
- …

Ik moet een rol/playbook schrijven om dit project zo kant en klaar mogelijk uit te rollen naar een IAAS cloud-provider of een andere virtuele machine. De bedoeling hiervan is om de oplossing aan te beiden zodat iemand gemakkelijk een Linux VM kan deployen, de playbook met de juiste variabelen start en dan alles kant en klaar geconfigureerd is.

Key attributen

- Verschillende scans
- Verschillende tests/checks
- Opensource technologieën
- Ansible playbook
- Gebruiker krijgt telegram berichten

Functionele analyse

Hier beschrijf ik de stappen die ik moet nemen om dit project tot werking te krijgen. Ik wil in stappen werken waarbij als het blijkt dat ik niet genoeg tijd heb om alles te doen, ik nog steeds een mooi een goed project heb.

Stap 1

Stap 1 bestaat uit de test omgevingen opzetten, Het is opgelegd om een basis infrastructuur van Rocky Linux 9 te gebruiken. Er wordt een playbook voorzien aan ons door de lector om deze op te zetten. Ik zal ook een lokale machine aanmaken om tests uit te voeren. Tijdens stap 1 ga ik ook onderzoeken wat er allemaal moet gebruiken en als de technologieën die ik wil gebruiken compatible zijn met Rocky Linux 9.

Stap 2

In stap 2 ga ik met de eerste tests beginnen. Deze test bestaat uit een virus scanner die een bepaalde directory scant op de machine elke keer als er een nieuwe file wordt in toegevoegd. Zo kan de gebruiker files naar deze directory sturen en zien als de malware wordt gedetecteerd of niet. Der resultaten hiervan worden naar de gebruiken gestuurd op telegram.

Deze test kan handig zijn voor verschillende redenen. Als je wilt zien als een verdachte file malware is of niet, of als je eventueel malware programma's zelf schrijft en je wilt weten als die gedetecteerd wordt of niet. Een andere reden om deze test te gebruiken is bijvoorbeeld als je een (niet malware) programma schrijft en je wilt zien als deze wordt gezien als malware of niet.

Deze test heb ik al geschreven op een Ubuntu VM. Hij werkt goed maar heeft meer optimalisatie nodig. Ik voer hem uit met een Docker Compose. Hij gebruikt de open source virus scanner, ClamAV www.clamav.net.

Stap 3

Ik heb het idee gekregen om na elke test die ik maak, om het werkende te krijgen in mijn Ansible script. Met deze aanpak heb ik na elk deel een afgewerkt product en kan ik met een gerust hart verder werken aan de andere testen.

Daarom ga ik in stap 3 mijn virus scan test in een Ansible script integreren.

Stap 4

In stap 4 ontwikkel ik een integriteit check met Hashlib. Het idee is om hashs van de files te berekenen en ze te vergelijken. Ik werk hierbij in stappen, eerst bereken ik de hash van elke file en slaag ik ze op. Na een verloop van tijd bereken ik de hashs opnieuw en vergelijk ik ze met de eerder opgeslagen hashs, als ze niet gelijk zijn weet ik dat de files aangepast of corrupt is en kan ik de gebruiker dit laten weten via een telegram bericht, of iets anders. Dit proces voert zich vanzelf terug uit na een ingestelde tijd. De files die gecheckt moeten worden zullen zich in een speciale directory bevinden.

Na deze test te schrijven integreer ik hem in mijn Ansible playbook.

Overige stappen

Na de eerste twee tests te maken zou ik normaal gezien een goed idee moeten hebben van hoe alles werkt, van mijn skills en mogelijkheden. Ik denk dat als ik het soort stappenplan aanpakt van, eerst mijn test maken en testen, en dan dit ga toevoegen in mijn Ansible playbook, dat ik grotendeels alles dat ik wil kan afkrijgen.

GIT

Ik zal Git gebruiken voor het versiebeheer en het inleveren van het project aan de lector. Ik ga een GitHub repository opzetten om daar in mijn code te beheren. Deze is privé en mag enkel door mij en de lector bezocht zijn.

Conclusie

Uit deze analyse heb ik gemerkt dat ik een goed idee heb van wat ik wil en ga doen. Ik heb veel zin in dit project en kijk er naar uit om hiermee in gang te schieten. Ik vind het leuk omdat ik persoonlijk een project was aan het maken hierrond en ik dit nu kan combineren met het project van Infrastructure as Code.

Ik denk dat mijn ideeën misschien niet het meest interessant of nuttig is, maar ik zal er extreem veel van bijleren en mijn kennis over Security en Cloud verbreden.