Building Red Team Infrastructure with Terraform

Moses Frost

WHO AMI?

- Moses Frost
- Red Team Operator {@} Neuvik
- SANS SEC588 Cloud Penetration Testing Author
- @mosesrenegade

Not an evil QR Code

http://bit.ly/m/mosesrenegade

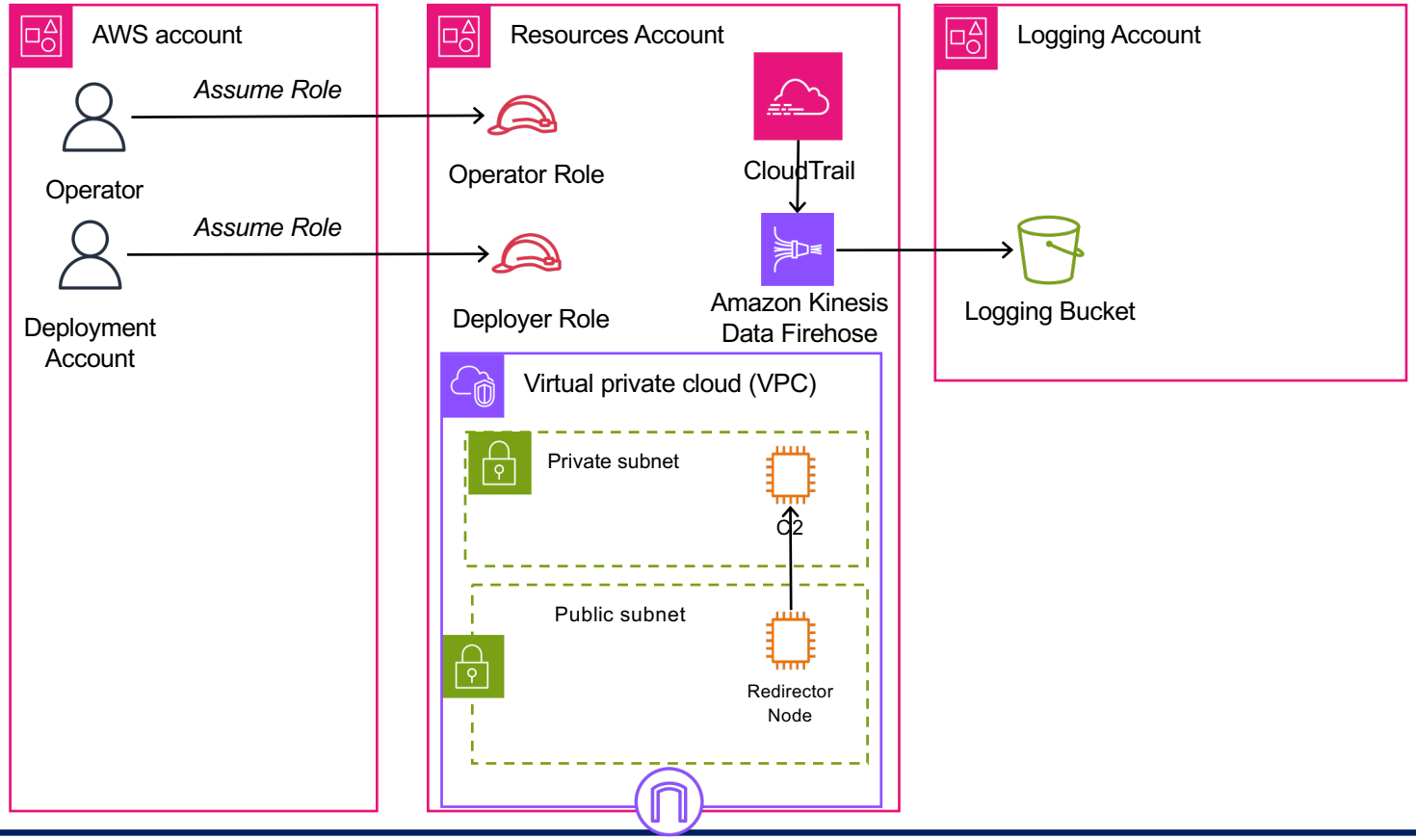
Todays Workshop Agenda

- Red Team Architectures are fairly complex
- The Cloud Offers tons of benefits:
  - Extensibility
  - Decomposability
  - Stealth
- Through automation you can have consistent architectures
  - This is a two-hour workshop, the idea is to get you going, but how far can you take it?
  - Will not be extremely comprehensive but will get you started.

Todays Workshop Agenda

- Background for today's Workshop
- Neuvik doesn't advertise this, but we have created automated Red Team Infrastructure for Clients
  - We have also used this to build our modular infrastructure
  - This will eventually be a Neuvik Infrastructure Master Class

Architecture Reference Example



Terraform

- Terraform has some benefits
- Provides for the skeletal framework (language and tooling) to deploy
  - **Terraform:** The toolset for building consistent day0 environments
  - **HCL:** HashiCorp Language
  - **Provider Modules:** Provider modules are per-platform, some are maintained by HashiCorp, others by individual contributors
    - **AWS Provider Module:** Used for building AWS Environments
    - **Azure Provider Module:** Used for building Azure Environments
- We will step through each piece in the lab.

Lab 1: Crash course intro

- The goal of this module:
  - Setup the terraform environment
  - Install the required binaries
    - Terraform from within the Terraform Website
    - tflint
  - Create Resources to start our build:
    - A VPC For our Resources
    - A Public Subnet for our resources
    - Internet Gateway and NAT Gateways
    - Routing Tables
  - We will also learn how to dynamically add variables between modules
  - Learning how to read provider documentation is essential

Lab 2: Building on Lab 1

- Building a Machine is critical, so how do we build a machine in AWS
  - Similar mechanisms for all the infrastructure cloud providers
- The goal of this module:
  - Build upon the first lab
  - Add a basic machine
  - Use destroy to build and rebuild
  - Use the Data Elements to pull in data

Lab 3: Fundamentals in Terraform

- Builds upon Lab 2
- The goal of this module:
  - What are variables?
    - How can we use variables and start reusing code blocks
  - How do you output what you build?
    - Output vs Show will be shown

Lab 4: What is Cloud Init?

- Builds upon Lab 3
- The goal of this module:
  - Using templates
  - Using Cloud-Init to build machines with our server software
    - You want a C2 server dynamically? How can we do that?
    - How do you want to do For – Loops?
    - Looping Over Keys and Values?

Lab 5: Getting Very Advanced

- Code Re-usability with Modules
  - What is a module?
  - How do I build and use a module?
  - What makes it re-usable
  - What about tainting and maintaining state?
  - How do I prevent accidental deletion?

URLS

- https://www.github.com/neuvik/neuvik-terraform-workshop
- https://bit.ly/terraform-workshop-url

workshopuser3

1 – cd Directory

2 – Copy the export

3 – terraform init

4 – git pull (to get data.tf)

5 – terraform fmt
6 – terraform plan -out "run.plan"
7 – terraform apply "run.plan"