Azure Technical Challenge

Welcome to Our Technical Challenge!

I'm glad you're here and excited to see what you bring to the table.

Our goal is to make the hiring process a true reflection of your skills while respecting your time. Rather than multiple technical interviews or live coding sessions, this take-home challenge lets you work at your own pace, in your own environment, and show us what you can really do.

Here's the deal:

- We expect this challenge to take a few hours If you want to dive deeper, explore more features, or fine-tune your solution, feel free—but it's not expected. We value work-life balance, and we're not here to overwhelm you.
- This challenge is meant to give you a representation of what we actually work on. No trick questions, no gotchas, just a real-world problem you can dig into.

What we're looking for:

- How you think through and solve problems.
- Your technical skills, including code quality and approach.
- Communication and clarity in how you present your work, whether in code or supporting notes.

When you're done, send us a link to your GitHub repository so we can dive into your solution and learn more about how you work.

If you run into any questions or need anything clarified along the way, don't hesitate to reach out.

We can't wait to see what you come up with!

Douglas Francis

Cloud Services Managing Principal

Challenge Overview

Create a proof-of-concept Azure environment using Terraform. The environment will host a basic web server with proper network segmentation and security controls. Use Coalfire's open source terraform modules as much as possible in your solution (https://github.com/orgs/Coalfire-CF/repositories?q=visibility:public+terraform-azure) The use of Terraform modules is required, but it is not required to use ours.

Your challenge when submitted should be all within a public GitHub repository, with your code, a diagram that depicts your solution and any notes you have from going through the challenge. The main repo README should document your solution, provide deployment steps and your notes and commentary from going through the challenge.

Any detail not provided in the scenario or requirements is up to your discretion.

Technical requirements

Network

- 1 VNet – 10.0.0.0/16
- 4 Subnets
    - Application, Management, Backend, Web. All /24

Compute

- 2 Virtual Machine in an availability set running Linux (your choice of distro) in the web subnet
    - NSG allows SSH from management VM, allows web traffic from the Load Balancer. No external traffic
    - Script the installation of Apache
- 1 Virtual Machine running Linux (your choice of distro) in the Management subnet
    - NSG allows SSH from a single specific IP or network space only

Supporting Infrastructure

- One storage account.
    - GRS Redundant
    - Only accessible to the VM in the Management subnet
    - One container “terraformstate”
    - One container “weblogs”
- One Load balancer that sends web traffic to the VM’s in the availability set.

Deliverables

1. Public GitHub repository containing:
    - All Terraform configurations
    - Architecture diagram
    - README including:
        - Solution overview
        - Deployment instructions
        - Design decisions and assumptions
        - References to resources used

  - Screenshot of successful SSH connection to management VM
  - Screenshots of Apache running on both AS VM's

Evaluation Criteria

We evaluate tech challenges based on

- Code Quality
  - Terraform best practices
  - Module usage
  - Correct resources deployed
- Security Implementation
  - Network segmentation
  - Access Controls
  - Azure best practices
- Architecture Design
  - Diagram clarity
  - Resource organization
- Documentation
  - Clear instructions
  - Well-documented assumptions
  - Proper reference citations
- Problem-Solving Approach
  - Solutions to challenges encountered
  - Design decisions

Guidelines

- Work independently – no collaboration
- We do encourage the use of web resources (Stack Overflow, Reddit, technical blogs, etc.) if used, provide links as part of your documentation.
- Document any assumptions and design decisions you make.
- Partial solutions with documented challenges are acceptable.
- Questions welcome – reach out if you need clarification