Welcome to the Wiz Technical Exercise! This is your opportunity to demonstrate compelling technical proficiency as well as your domain expertise in modern DevOps practices, cloud architecture, and cybersecurity. In this exercise you will **create and deploy** the environment, and then **deliver a presentation** on the exercise and environment. Your recruiter will provide you with a voucher code to leverage the Wiz CloudLabs platform for access to a cloud account.

Reach out to your hiring manager prior to completion to check-in to ensure you understand the expectations or clarify any parts of this exercise. READ THIS DOC COMPLETELY BEFORE STARTING.
The exercise can be deployed in AWS, Azure or GCP and will consist of:

1. Deploy a two-tier web application (front-end / database) following the requirements and using several cloud services which contain intentional configuration weaknesses
2. Leverage modern DevOps tools/best practices when possible to automate the building of your cloud infrastructure and application deployment
3. Implement technical security controls before your application reaches production. These controls may cover a variety of different categories, such as compliance, audit, or security monitoring.

When you are ready, we'll schedule you with a Wiz expert panel. Plan for your presentation and demo to take about 45 minutes, during which the panelist will ask questions about the environment as well. With the remaining 15 minutes, we can have a discussion about how Wiz works and answer any questions you may have about our solution.

The Presentation

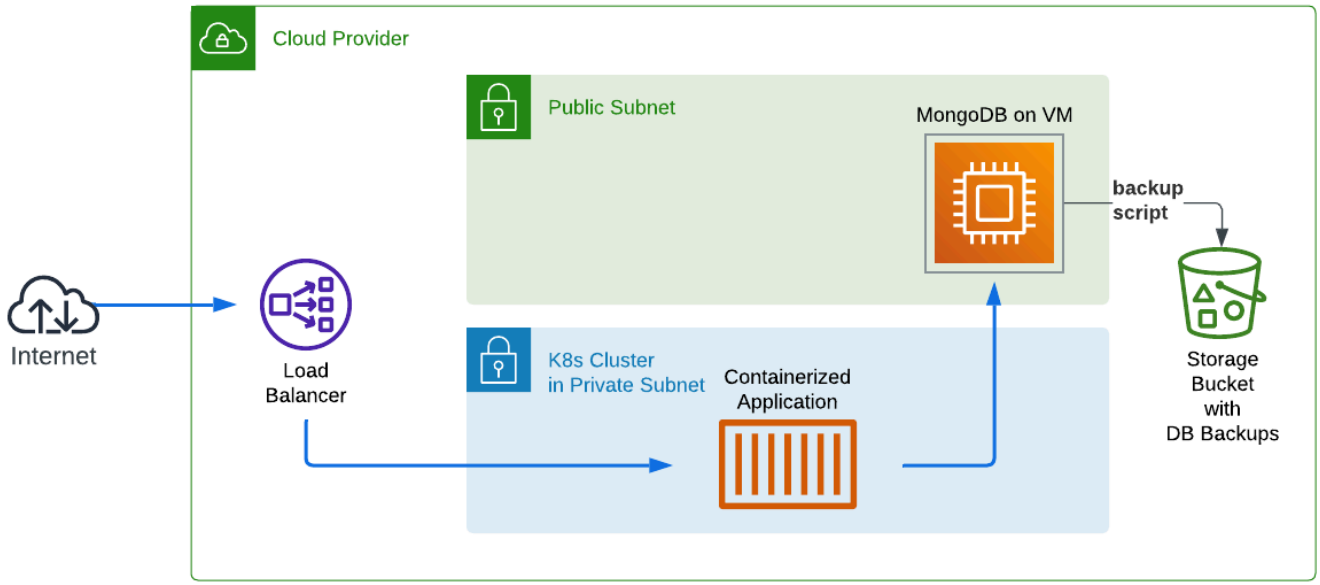
You will be presenting your solution (leveraging the provided template) to 2-3 Wiz panelists via Zoom. The panel will assess what you built, your methodology, challenges faced, your insights on the environment's security & the quality of your overall presentation. Your approach, the challenges faced, and your solutions will be essential components for the presentation portion of this task, along with your ability to effectively communicate the technical details.

Excellent presentations will:

- Incorporate a mix of slides and live walkthrough
- Evidence the proper function of the infrastructure
- Discuss your approach to the build-out, including challenges and adaptations
- Detail weak configurations and their potential consequences *(optional for support roles)*
- Demonstrate the value of the security tool in reducing risk *(optional for support roles)*

The WebApp Environment

The following diagram depicts the environment you will be building. You must use a containerized web application in which you build the container image, and expose this via a load balancer to the public internet. You must deploy a virtual machine running an outdated version of MongoDB, in which this MongoDB instance must be leveraged by your containerized application. The MongoDB database must be backed up into a cloud storage bucket. You may use IaC to complete this section too.



Virtual Machine with Mongo Database Server

This database server must be leveraged by the web application. The database backups must be automated and stored in the public-readable cloud object storage.

- VM should be leveraging a 1+ year outdated version of Linux
  - SSH must be exposed to the public internet
  - VM should be granted overly permissive CSP permissions (e.g. able to create VMs)
- Database should be MongoDB that is a 1+ year outdated database version
  - Access must be restricted to Kubernetes network access only and require database authentication
- Database must be automatically backed up on a daily basis to a cloud object storage
  - Object storage must allow public read and public listing

Web Application on Kubernetes

The web application must be a containerized application for which you have (re-) built the container image. The application can be your own choice, but must leverage the MongoDB database. A sample todo list application is available here.

- Kubernetes cluster must be deployed in a private subnet

- Access to MongoDB must be configured via an environment variable configured in Kubernetes
- The container image must contain a file called **wizexercise.txt** and contain your name
    - You must provide in the exercise how you got the file in and validate the file actually exists in the running container image
- Container application must be assigned a cluster-wide kubernetes admin role and privilege
- Container application must be exposed via a Kubernetes ingress and CSP load balancer
- You must be able to demonstrate the Kubernetes CLI (kubectl) during your demonstration
- You must be able to demonstrate the web application and prove the data is in the database

Dev(Sec)Ops

For specific roles (refer to your hiring manager or recruiter for details), Wiz requires our Wizards to have deeper knowledge of VCS, SCM, and CI/CD pipelines. Implementing this will also simplify the overall setup and tear down of an environment like this. If not required for your role, completing this will be seen as a bonus.

- **VCS/SCM**: You must push your code & config to a VCS/SCM of your choice (GitHub, GitLab, ADO, etc.)
- **CI/CD Pipelines:** You must setup at least two pipelines:
    - One CI/CD pipeline to securely deploy this exercise as Infrastructure-as-Code (IaC)
    - One CI/CD pipeline to build & push the application as a container to a registry and trigger a Kubernetes deployment of the container image.
- **Pipeline Security:** You must implement security controls in your VCS platform both for the repository as well as scanning the IaC code + Container Image prior to deployment.
- *Optional* **Simulation:** You may run a simulated attack or behaviors in the cloud environment and pipelines to showcase the efficacy of your preventative and detective security controls.

Cloud Native Security

For specific roles (refer to your hiring manager or recruiter for details), you must implement native CSP tooling to detect the misconfigurations in the environment. You will demonstrate this tooling in the demo portion of the presentation.

- You must configure control plane audit logging for your CSP environment
- You must implement at least one preventative cloud control
- You should implement at least one detective cloud control
- You must demonstrate these tools and their impact
- *Optional* In addition to the above, you can implement & demonstrate security in the CI/CD phases