

Wiz Field Technical Exercise

Objective:

Demonstrate your proficiency in deploying and managing cloud-based solutions by setting up a three-tiered web application in the cloud platform of your choice.

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.

You can choose the cloud environment of your preference, but we encourage you to pick an environment that isn't your strongest. This offers a better insight into your problem-solving and adaptability skills.

Environment Requirements:

Web Application Tier: A containerized web application deployed in a Kubernetes cluster.

Database Tier: MongoDB server set up on a VM configured for access by the Kubernetes cluster.

Storage Tier: Object storage configured to hold MongoDB backups and set with public read permissions.

Build Out Overview:

Your web application will be publicly accessible and is set up to communicate with the MongoDB server. The MongoDB database backups should be scripted and stored into the public-read object storage.

Upon completing the exercise, we will schedule a technical interview panel where you'll have up to forty-five minutes to present your PowerPoint, demonstrate the setup (live), and for us to ask questions. Then either, we'll have a 30-minute demo of Wiz and you'll have the opportunity to ask us any questions or a separate call will be scheduled.

The panel will assess what you built, your methodology, challenges faced, and your insights on the environment's security.

Environment Components:

- A containerized web application deployed on a Kubernetes cluster and exposed through a cloud-native load balancer.
- The application should interact with the MongoDB instance.
- A publicly readable object storage bucket containing MongoDB backups.

Setup:

At the end of basic setup, you should have a **working web application** which can be accessed from the web, and a VM instance running MongoDB. Before the presentation begins, ensure that every component of the exercise is operational and ready for demonstration.

- **MongoDB:** Set up a VM using an outdated Linux version. On top of this Install an older MongoDB Major release package on this VM.
- **MongoDB Authentication:** Ensure MongoDB uses authentication so you can construct a MongoDB connection string.
- **Highly Privileged MongoDB VM:** Configure the VM in a way that it is granted Admin CSP permissions.
- **Object Storage:** Create a bucket which will store the MongoDB backups. Modify the permissions to allow public read access to it as well as to the backups stored within. *You will be asked to validate during the review that the backup is accessible via an external url.*
- **MongoDB Backups:** Create a Script which regularly backups the MongoDB databases and transfers them to the created bucket.
- **Kubernetes Cluster:** Deploy a managed Kubernetes cluster in the same network as the MongoDB VM.
- **Container Web Application:**
 - Deploy a containerized web application to the Kubernetes cluster. You can develop your own, utilize open-source solutions, or try this sample: <https://github.com/jeffthorne/tasky>
 - Ensure the container employs the MongoDB authentication. This typically uses a connection string format.
 - Confirm the built container image includes a file named "wizexercise.txt" with content.
- **Public Access:** Set up routing to provide public access to the Kubernetes cluster, typically using a load balancer.
- **Container Admin Configuration:** Provide the container with cluster-admin privileges as described here in the ["Permissive RBAC permissions" section](#)

- **Bonus:** Deploy Everything using Infrastructure-as-Code

Additional Resources:

- Identity and Access Management Policies:
 - AWS Identity and Access Management (IAM) Documentation: [What is IAM?](#)
 - Google Cloud IAM Documentation: [Cloud Identity and Access Management \(IAM\)](#)
 - Azure Identity and Access Management: [Azure Active Directory Documentation](#)
- Infrastructure-as-Code tooling for Kubernetes deployment:
 - Terraform Kubernetes Provider: [Terraform Kubernetes Provider Documentation](#)
 - Helm – The Kubernetes Package Manager: [Helm Official Site](#)