

# **Technical Presentation: Running Doom in UpCloud Kubernetes Service (UKS)**

Presented by Taurai Mutimutema

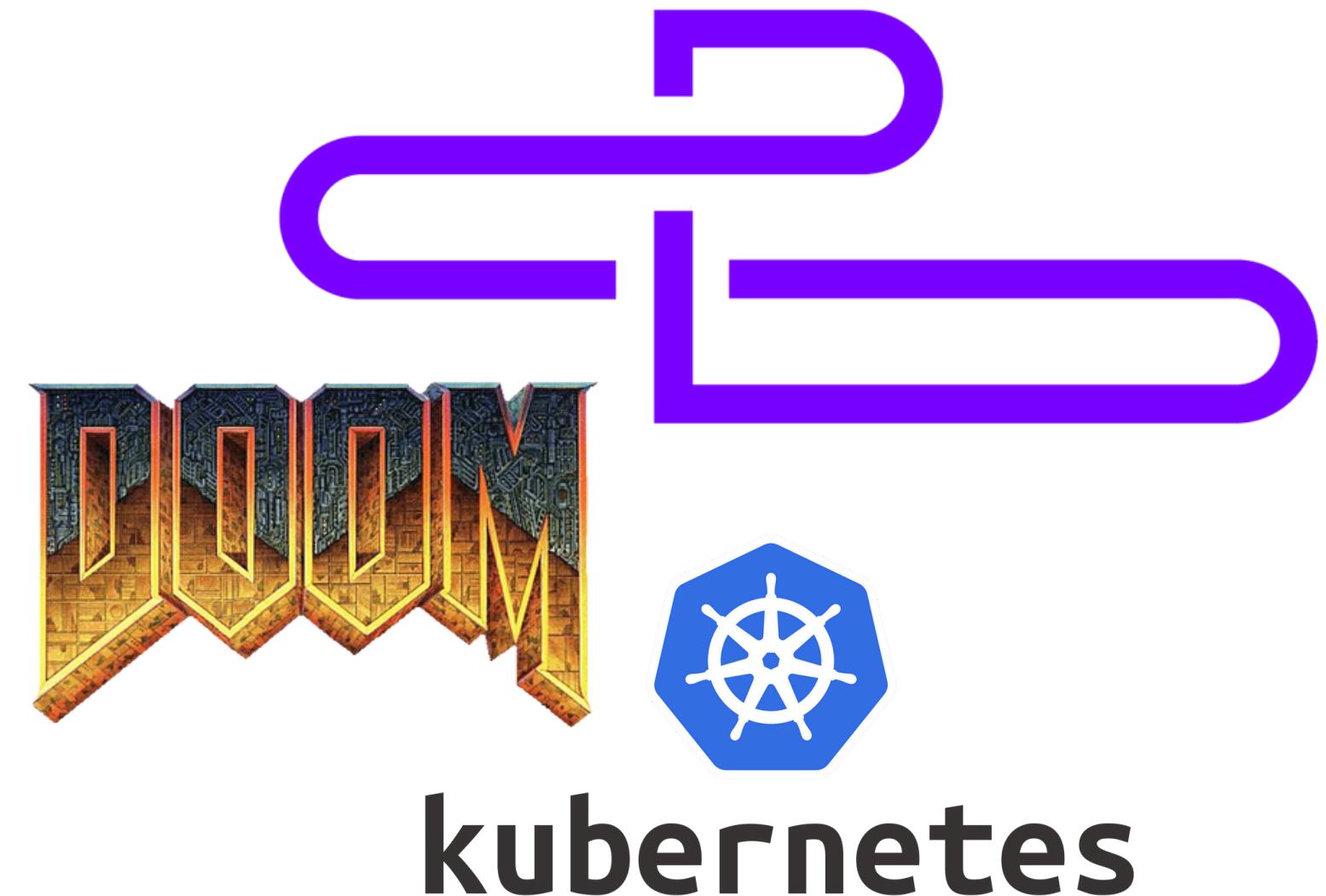
Date: 28 May 2025

# Introduction

This presentation deck is a piece of work in response to the Developer Advocate role interview process requirement to demonstrate technical aptitude.

Specifically, "Yeah, but does it run Doom? Running Doom in UpCloud Kubernetes Service."

A Technical Demonstration of KubeDoom on UpCloud





# About Me

**2010 - 2015** ●

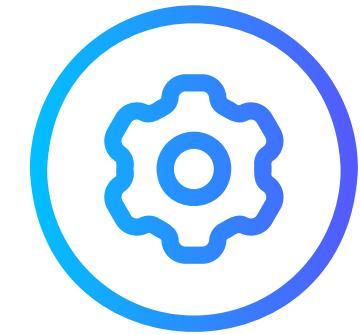
*Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque facilisis tincidunt risus sit amet sagittis. Praesent tempus vitae felis maximus pretium. Mauris quam dui, scelerisque.*

# Objectives

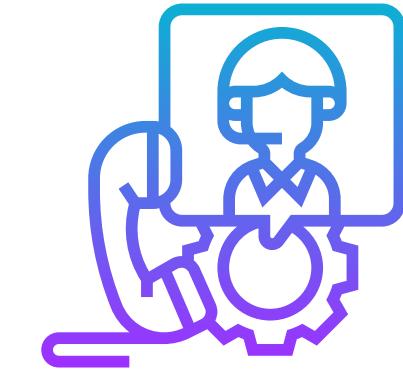
Run the Doom game on the UpCloud Kubernetes service.



**Research and Plan**



**Test and Execute**



**Present a Demo**

# The Process



## Account Creation

- Log in to UpCloud Control Panel
- Navigate to Kubernetes section
- Create a new cluster with appropriate specifications
- Download kubeconfig file
- Configure kubectl to use the cluster



## Provisioning

- Create a namespace for isolation
- Set up service account & RBAC permissions
- Prepare deployment manifest
- Use kustomization to provision



## Building

- namespace.yaml
- rbac.yaml
- deployment.yaml
- kustomization.yaml



## Playing

- Connect using any VNC client
- Address: localhost:5900
- Password: \*\*\*\*\*
- Game controls:
- Arrow keys: Move
- Ctrl: Shoot
- Space: Open doors
- Esc: Pause

# Problems



## Problems

- Unfamiliarity with the UpCloud platform in general. Interface, CLI commands, and basic tier limitations.
- System compatibility issues (Go on M4)



## Solutions

- Took the weekend to familiarize myself with the platform (read the docs and tested a few simple deployments with free-trial accounts)
- Reverted to using kubectl instead of upctl due to system issues when installing the supported version of Go on M4 MacBook.

# What is KubeDoom?

- A visualization tool for Kubernetes resources
- Pods appear as monsters in the game
- Shooting a monster deletes the corresponding pod
- Based on the original Doom game from 1993
- Educational and entertaining way to interact with Kubernetes

## Notes:

KubeDoom is a creative visualization tool that represents your Kubernetes pods as monsters in the classic Doom game. When you shoot a monster in the game, it actually deletes the corresponding pod in your Kubernetes cluster.

It's based on the original Doom game from 1993 and provides an educational yet entertaining way to interact with your Kubernetes resources. It's a perfect example of how we can make DevOps more engaging.



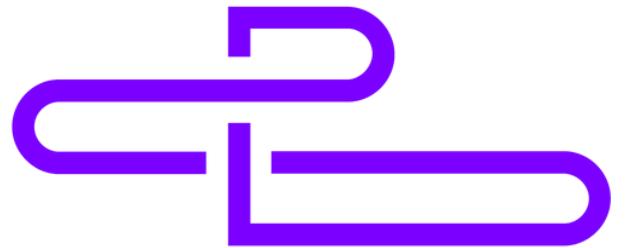
# Prerequisites

- UpCloud account with credits
- UKS cluster created and running
- upctl or kubectl configured to access your UKS cluster
- VNC Viewer installed on your local machine

Special thanks go to the UpCloud team for providing test credits for this project to commence.

K8 knowledge is assumed.

I went with an open-source VNC Viewer project, Remote Ripple.



kubernetes



# Config files

namespace.yaml

```
! namespace.yaml
1  apiVersion: v1
2  kind: Namespace
3  metadata:
4    name: kubedoom
5
```

rbac.yaml

```
! rbac.yaml
1  apiVersion: rbac.authorization.k8s.io/v1
2  kind: ClusterRoleBinding
3  metadata:
4    name: kubedoom
5  roleRef:
6    apiGroup: rbac.authorization.k8s.io
7    kind: ClusterRole
8    name: cluster-admin
9  subjects:
10 - kind: ServiceAccount
11   name: kubedoom
12   namespace: kubedoom
```

serviceaccount.yaml

```
! serviceaccount.yaml
1  apiVersion: v1
2  kind: ServiceAccount
3  metadata:
4    name: kubedoom
5    namespace: kubedoom
```

kustomization.yaml

```
! kustomization.yaml
1  apiVersion: kustomize.config.k8s.io/v1beta1
2  kind: Kustomization
3  namespace: kubedoom
4  resources:
5  - namespace.yaml
6  - serviceaccount.yaml
7  - rbac.yaml
8  - deployment.yaml
```

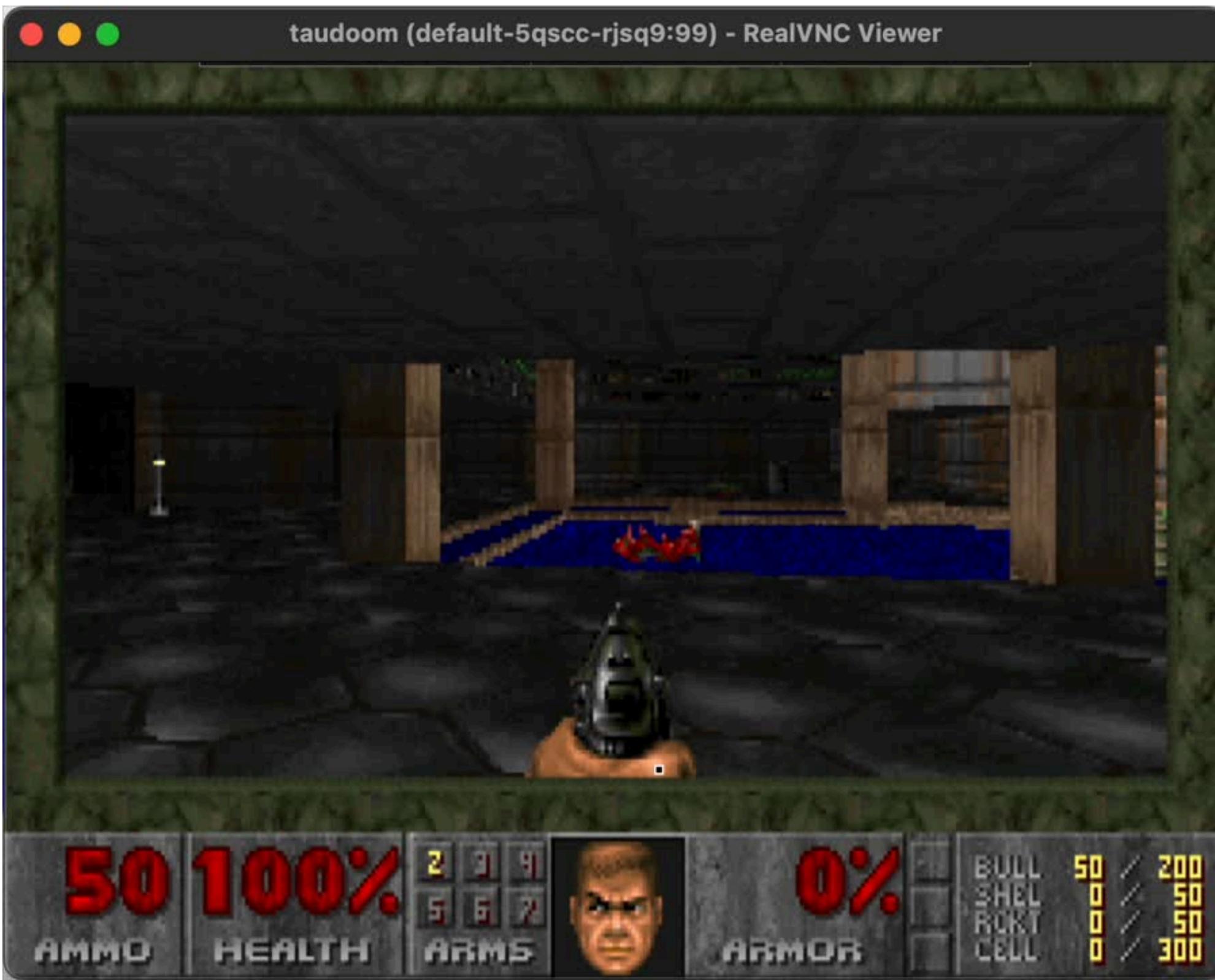
## deployment.yaml

```
! deployment.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    labels:
5      app: kubedoom
6      name: kubedoom
7      namespace: kubedoom
8  spec:
9    replicas: 1
10   selector:
11     matchLabels:
12       app: kubedoom
13   template:
14     metadata:
15       labels:
16         app: kubedoom
17     spec:
18       hostNetwork: true
19       serviceAccountName: kubedoom
20       containers:
21         - image: ghcr.io/storax/kubedoom:latest
22           name: kubedoom
23           env:
24             - name: NAMESPACE
25               value: default
26           ports:
27             - containerPort: 5900
28               name: vnc
```

# Lab Walkthrough

- Refer to detailed walkthrough of the process in GitHub Docs  
[docs/Lab Guide\\_ Running Doom in UpCloud Kubernetes Service \(UKS\).md](#)

# Gameplay Demo



# Thank you

## Contact Me

**Tau. M**

Developer Advocate

 +48 532 007 086

 rusiqe@gmail.com

