Platomics GmbH
www. platiomics.com
Jakov-Lind-Strasse 15/3 | 1020 Vienna | Austria

Homework Assignment

The homework consists of two parts that are tied together, but one can be solved without completing the other.

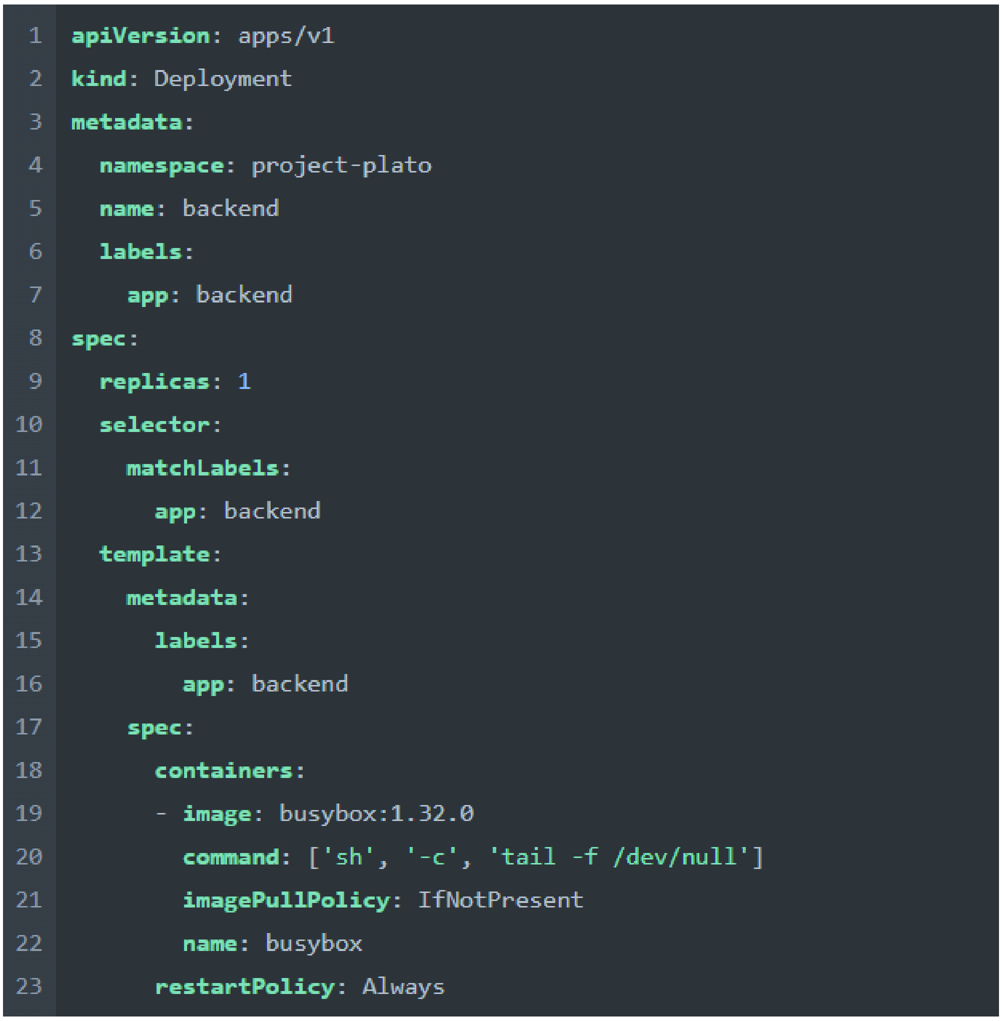
Once finished, please provide the code in a public Git repository (eg.: GitHub/GitLab) and send the link to Dominik (dominik.luntzer@platomics.com) and Simone(simone.grossi@platomics.com).

Part 1 - Kubernetes workloads Challenge

Your task is to create Kubernetes manifests according to the task's requirements. The results of the task should be easily reproducible, therefore provide documentation on how to verify the results of your work, e.g. launching a cluster, verifying pod readiness etc.

Requirements:

1. Create an immutable Deployment called backend using this base manifest:



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  namespace: project-plato
  name: backend
  labels:
    app: backend
spec:
  replicas: 1
  selector:
    matchLabels:
      app: backend
  template:
    metadata:
      labels:
        app: backend
    spec:
      containers:
      - image: busybox:1.32.0
        command: ['sh', '-c', 'tail -f /dev/null']
        imagePullPolicy: IfNotPresent
        name: busybox
      restartPolicy: Always
```

2. Modify the *Deployment* in a way that no processes inside the container can modify the local filesystem, `/tmp` directory should be writeable. Don't modify the container image.

3. Create 2 more Deployments named `db1` and `db2` with image `nginx:1.16.1-alpine` in the same namespace.
4. Expose the `db*` Deployments within the cluster:
    a. `db1` via port 6379
    b. `db2` via port 5432
5. For `backend`, configure a *LivenessProbe* which simply executes command `true`.
   Also configure a *ReadinessProbe* which checks whether the `db1` is reachable via port 6379.
   → Confirm the functionality of the ReadinessProbe.
6. To prevent access to the whole cluster from a single compromised `backend` Pod, create a *NetworkPolicy* called `np-backend` in *Namespace* `project-plato`. It should allow the `backend-*` *Pods* only to:
    a. connect to `db1-*` Pods on port 6379
    b. connect to `db2-*` Pods on port 5432
   → Confirm the functionality of the newly created *NetworkPolicy*.
7. Create a *Secret* containing a username and a password and make it available inside the `db2` deployment.
   → Confirm the availability of the *Secret* contents in the *Deployment*.

Bonus task:

Deploy 'Postgres' using its respective `helm` *chart;*

Deploy 'kube-prometheus-stack' using its respective `helm` *chart;*

Configure both releases in a way that Postgres metrics can be queried in the Prometheus UI.

Part 2 - Infrastructure

Your task is to create a plan to make your workloads available to users publicly in a production environment. Prioritize security and collaboration ability of developers, i.e. the workloads will be running containers with applications developed internally at Platomics.

Requirements:

- Developers should ideally be able to self-manage the production deployment of their applications; Mention tools/services you can provide to enhance the productivity and satisfaction of our developers.
- Mention what technologies you would use to publish and deploy the Kubernetes workloads; mention additional services to be deployed that could reinforce security, observability etc., if any.
- Create a simple diagram to illustrate your proposed architecture. The goal is clarity, not complexity. Ensure a teammate could easily understand it. Use a tool like http://diagrams.net or similar.

You can use any technologies you like, but prioritize those mentioned in the job advertisement if applicable.

You are **NOT** required to implement the architecture. This is a design exercise.

Bonus task:

Expand on the example from task 1 by (partially) implementing your proposed architecture 😉