Lab 9: Serving ML with Kubernetes

In this lab, we'll deploy the churn preduction model from earlier. We already have a docker image for this model - we'll use it for deploying the model to Kubernetes.

Bulding the image

Clone the course repo if you haven't:

```

git clone https://github.com/fenago/mlbookcamp-code.git
```

Go to the `course-zoomcamp/05-deployment/code` folder and execute the following:

```bash
docker build -t churn-model:v001 .
```

> **Note:** If you have troubles building the image, you can
> use the image I built and published to docker hub:
> `fenago/zoomcamp-model:churn-v001`

Run it to test that it's working locally:

```bash
docker run -it --rm -p 9696:9696 churn-model:v001
```

And in another terminal, execute `predict-test.py` file:

```bash
python predict-test.py
```

You should see this:

```
{'churn': False, 'churn_probability': 0.3257561103397851}
not sending promo email to xyz-123
```

Now you can stop the container running in Docker.

Installing `kubectl` and `kind`

You need to install:

* `kubectl` - https://kubernetes.io/docs/tasks/tools/ (you might already have it - check before installing)
* `kind` - https://kind.sigs.k8s.io/docs/user/quick-start/

Quesion 1: Version of kind

What's the version of `kind` that you have?

Use `kind --version` to find out.

Creating a cluster

Now let's create a cluster with `kind`:

```bash
kind create cluster
```

Question 2: Verifying that everything works

Now let's test if everything works. Use `kubectl` to get the list of running services.

What's `CLUSTER-IP` of the service that is already running there?

Question 3: Uploading the image to kind

To be able to use the docker image we previously created (`churn-model:v001`), we need to register it with kind.

What's the command we need to run for that?

Question 4: Creating a deployment

Now let's create a deployment (e.g. `deployment.yaml`):

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: churn
spec:
  selector:
    matchLabels:
      app: churn
  template:
    metadata:
      labels:
        app: churn
    spec:
      containers:
      - name: churn
        image: <Image>
        resources:
          limits:
            memory: "128Mi"
            cpu: "500m"
        ports:
        - containerPort: <Port>
```

Replace `<Image>` and `<Port>` with the correct values.

What is the value for `<Port>`?

Question 5: Pod name

Apply this deployment:

```yaml
kubectl apply -f deployment.yaml
```

Now get a list of running pods.
What's the name of the pod that just started?

Question 6: Creating a service

Let's create a service for this deployment (`service.yaml`):

```yaml
apiVersion: v1
kind: Service
metadata:
  name: <Service name>
spec:
  type: LoadBalancer
  selector:
    app: <???>
  ports:
  - port: 80
    targetPort: <PORT>
```

Fill it in. What do we need to write instead of `<???>`?

Apply this config file.

Testing the service locally

We can do it by forwarding the 9696 port on our computer to the port 80 on the service:

```bash
kubectl port-forward service/churn 9696:80
```

Run `predict-test.py` from session 5 to verify that everything is working.