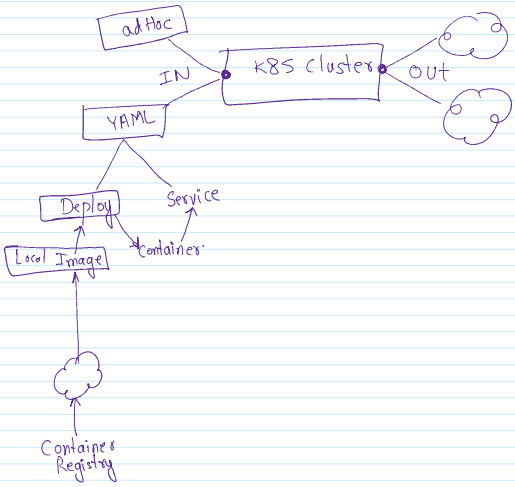
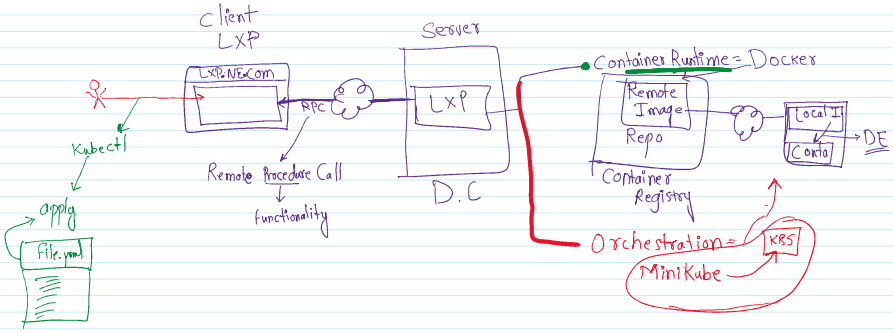
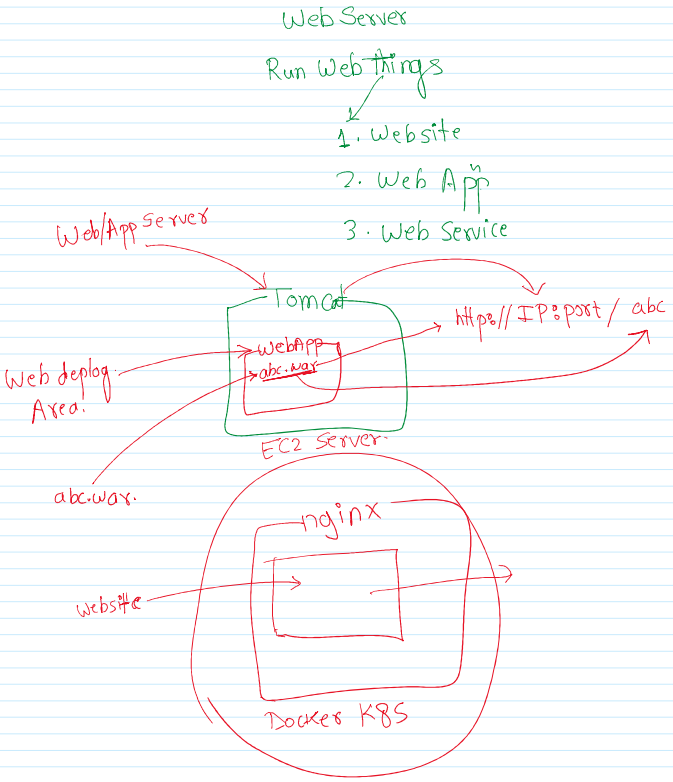
38.18May.K8SConcepts-Implementations

Thursday, May 18, 2023 7:07 PM








```
apiVersion: v1
kind: Pod

metadata:
    name: webserver
```

```
    spec:
        containers:
        - name: webserver
            image: nginx:latest
            ports:
            - containerPort: 80
```

```
docker --version
docker info
kubectl version
minikube status
```

```
apiVersion: v1
kind: Service
metadata:
  name: helloworld
  labels:
    app: helloworld
spec:
  ports:
  - name: http
    port: 8000
    targetPort: 8000
  selector:
    app: helloworld
  type: LoadBalancer
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: helloworld
  labels:
    app: helloworld
spec:
  replicas: 2
  selector:
    matchLabels:
      app: helloworld
  template:
    metadata:
      labels:
        app: helloworld
    spec:
      containers:
      - name: helloworld
        image: signalsciences/example-helloworld:latest
        imagePullPolicy: IfNotPresent
        args:
        # Address for the app to listen on
        - localhost:8000
        ports:
        - containerPort: 8000
```

```
apiVersion: v1      # apiVersion: this is the version of the API used by the cluster.
                    # With new versions of Kubernetes being released, new functionality is introduced and, hence, new API versions may be defined.
                    # For the pod object, we use API version v1.
kind: Pod

metadata:           # Metadata: here we can define data about the object we are about to create.
  name: webserver   # In this example, we only provide the name of the pod. But you can provide other details like the namespace.
spec:               #The spec part defines the characteristics that a given Kubernetes object should have.
                    # It is the cluster's responsibility to update the status of the object to always match the desired configuration.
                    # In our example, the spec instructs that this object (the pod) should have one container with some attributes.
  containers:
  - name: webserver  # The name that this container will have.
    image: nginx:latest # The image on which it is based.
    ports:              # The port(s) that will be open.
    - containerPort: 80
```