Steps to Deploy Zipkin on Kubernetes and Expose via Node Port

Step 1: Create a Namespace for Zipkin (Optional)

It's a good practice to create a separate namespace for organizational purposes.

```
kubectl create namespace zipkin
```

Step 2: Create a Deployment for Zipkin

Create a YAML file named `zipkin-deployment.yaml` with the following content. This file defines a Deployment for Zipkin, pulling the official Docker image from Docker Hub.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: zipkin
  namespace: zipkin
spec:
  replicas: 1
  selector:
    matchLabels:
      app: zipkin
  template:
    metadata:
      labels:
        app: zipkin
    spec:
      containers:
      - name: zipkin
        image: openzipkin/zipkin:latest
        ports:
        - containerPort: 9411
```

Apply the deployment:

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

Step 3: Expose the Zipkin Service via NodePort

Now, create a Service to expose Zipkin externally via a NodePort.

Create a YAML file named `zipkin-service.yaml` with the following configuration:

```
apiVersion: v1
kind: Service
metadata:
  name: zipkin
  namespace: zipkin
```

```
spec:
  type: NodePort
  selector:
    app: zipkin
  ports:
    - port: 9411          # Port on the Service
      targetPort: 9411     # Port on the Zipkin container
      nodePort: 30000      # NodePort (choose a port between 30000-32767)
```

Apply the service configuration:

```
kubectl apply -f zipkin-service.yaml
```

Step 4: Verify the Deployment and Service

Check if the Zipkin pod is running:

```
kubectl get pods -n zipkin
```

Check if the service is available:

```
kubectl get services -n zipkin
```

Step 5: Access the Zipkin UI

Now, Zipkin should be accessible on any node IP address in your cluster at the specified NodePort. Open a web browser and go to:

```
http://<NODE_IP>:30000
```

This will open the Zipkin UI, where you can start viewing and analyzing tracing data as it's collected from your microservices.