# Application deployment and telemetry data

Tasks:

1. Install JDK8 or JDK11
2. Install build tool - maven (maven.apache.org)
3. Build all (script-build-components.sh)
4. Publish demo application to the DockerHub public repository (see script-publish-components.sh)
5. Deploy demo application to local Kubernetes
   1. Create ingress controller: /gateway/course-ingress-controller.yml
   2. Create gateway: /gateway/course-gateway-controller.yml
   3. Deploy frontend service: /frontend/deployment/deployment.yml
   4. Deploy books service: /books/deployment/deployment.yml
   5. Deploy authors service: /authors/deployment/deployment.yml
6. Execute query against application. Get ingress ip address: kubectl get ingress.
   1. http://[ingress api]/frontend-catalog/api/v1/dashboard
7. Open Kiali UI (login/password: admin/admin)
   1. istioctl dashboard kiali
8. Open Jaeger UI (find traces)
   1. istioctl dashboard jaeger
9. Open Grafana (use Istio Service Dashboard)
   1. Istioctl dashboard grafana

Checklist:

1. Kiali UI is accessible
2. Jaeger UI is accessible
3. Grafana UI is accessible
4. Application is deployed and accessible

Output:

Please create screenshots and commit them to the git repository under folder ServiceMesh\Session 2

1. Screenshot of the Kiali
   1. Application tab
   2. Application graph
2. Screenshot of the Jaeger
3. Screenshot of the Grafana
   1. Frontend application metrics
   2. Books application metrics
   3. Authors application metrics
4. Screenshot of the response of your application

Learning:

1. Istio telemetry (<https://istio.io/docs/tasks/telemetry/>)
2. Istio telemetry configuration (<https://istio.io/docs/reference/config/policy-and-telemetry/>)
3. Istio mixer component (<https://istio.io/docs/reference/config/policy-and-telemetry/mixer-overview/>)
4. Istio monitoring (<https://istio.io/docs/tasks/observability/metrics/using-istio-dashboard/>)
5. Kiali Tool: <https://kiali.io/documentation/architecture/>
6. OpenTracing: <https://opentracing.io/docs/overview/>