# Application deployment and telemetry data

Tasks:

1. Install JDK11 (npm)
2. Install a build tool - maven (maven.apache.org, skip if npm)
3. Build all services (scripts/build-components.sh, scripts/publish-components.sh)
4. Publish demo application to your DockerHub repository (see scripts/publish-components.sh) (optional)
5. Deploy demo applications to local Kubernetes
   1. Create an ingress controller: k8s/gateway/course-ingress-controller.yml
   2. Create a gateway: k8s/gateway/course-gateway-controller.yml
   3. Deploy a frontend service: k8s/deployments/frontend.yml
   4. Deploy a book service: k8s/deployments/book.yml
   5. Deploy an author service: k8s/deployments/author.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
5. **Optional. Install Kubernetes metrics**

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/observability/>)
2. Istio monitoring (<https://istio.io/docs/tasks/observability/metrics/using-istio-dashboard/>)
3. Kiali Tool: <https://kiali.io/docs/architecture/architecture/>
4. OpenTracing: <https://opentracing.io/docs/overview/>