The monitoring, logging, tracing and alerting is working. We want to focus in implementing SLOs and tuning our alerts.
All relevant resources are again located in https://gitlab.fhnw.ch/cloud/devops/templates/monitoring-logging-tracing and updated over there.

Task 1: Tweak your alerts

Your alerting is too verbose. The reason behind is the fact that the kube-prometheus-stack is designed to work on all kinds of cluster, even the onPrem and productive ones.
We want to reduce the noise.

Take a look at the alerts on prometheus itself (http://localhost:FORWARDEDPORT/alerts?search=) after forwarding prometheus (it is the Pod with port 9090 exposed) and determine, if the alert is necessary or not.

The alerts are described under https://runbooks.prometheus-operator.dev/runbooks/general . You can switch them off in the kube-prometheus-stack/values-devops.yaml . An example how to do it is in the template repository, check out the element "defaultRules".

Task 2: Define your SLO

Based on all monitoring metrics, you should be able to define a SLO. What could be a Service Level Indicators (SLI) of your chatbots-application? Define them for roberta and eliza.

Afterwards, based on the SLI, define your Service Level Objective.

If you need multiple queries, you can combine those as recording queries, the template under kube-prometheus-stack/values-devops.yaml is adapted with one example, search for the element "additionalPrometheusRulesMap".

If you need inspiration / background, feel free to read through https://github.com/seznam/slo-exporter/blob/master/docs/defining_new_slo.md , https://medium.com/@sklik.devops/our-journey-towards-slo-based-alerting-bd8bbe23c1d6 to learn how to implement them. You can keep it simple: Just define one SLO directly as alert in the prom-stack.