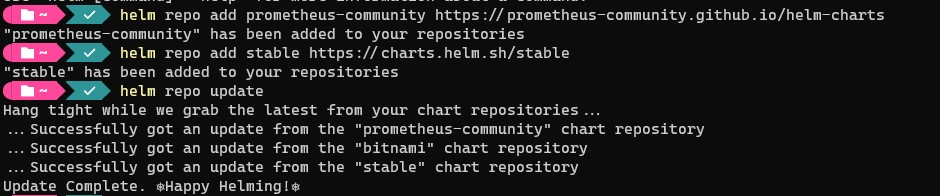
1. Add helm repo

helm repo add prometheus-community https://prometheus-community.github.io/helm-charts

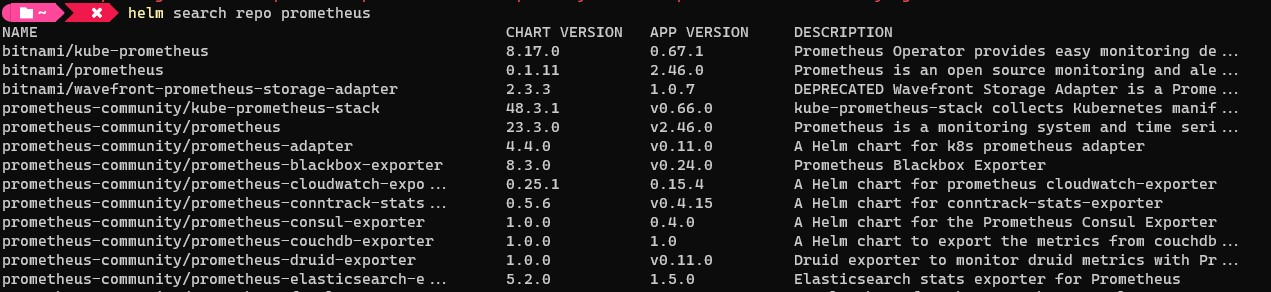
helm repo add stable https://charts.helm.sh/stable

helm repo update



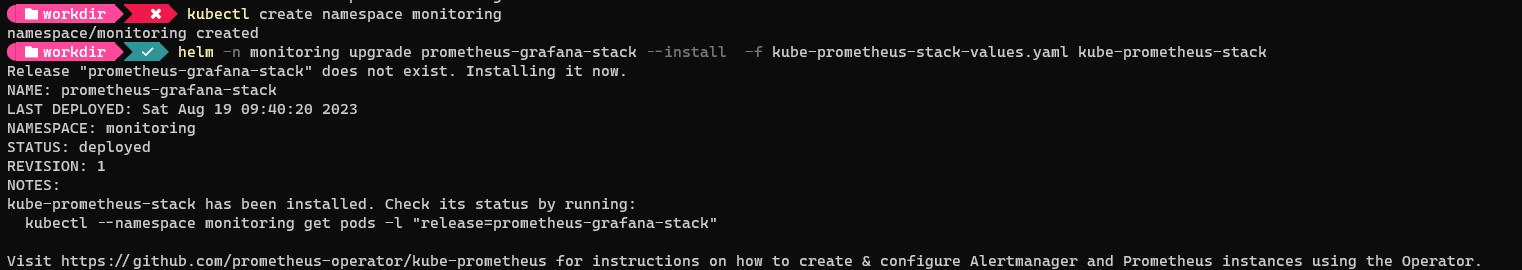
1. Search prometheus

helm search repo prometheus



1. Pull latest helm kube-prometheus-stack

helm pull prometheus-community/kube-prometheus-stack --version 48.3.1



1. Untar, copy "cp kube-prometheus-stack\values.yaml kube-prometheus-stack-values.yaml" and edit "code .\kube-prometheus-stack-values.yaml"

podMonitorSelectorNilUsesHelmValues: false

serviceMonitorSelectorNilUsesHelmValues: false

ruleSelectorNilUsesHelmValues: false

…..

additionalScrapeConfigs:

- job\_name: "backend"

scrape\_interval: 15s

metrics\_path: /api/metrics

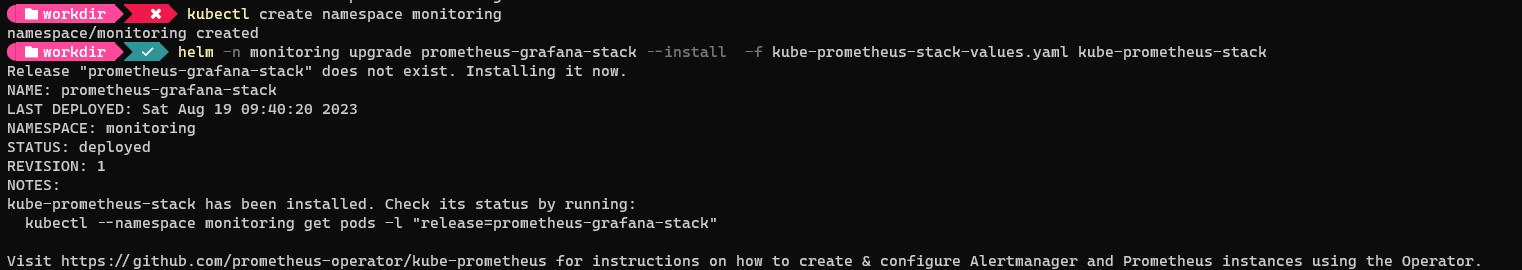
static\_configs:

- targets:

- backend.node.svc.cluster.local:3000

1. Install helm with our custom values

helm -n monitoring upgrade prometheus-grafana-stack --install -f kube-prometheus-stack-values.yaml kube-prometheus-stack



1. Confirm helm install successful

helm -n monitoring ls



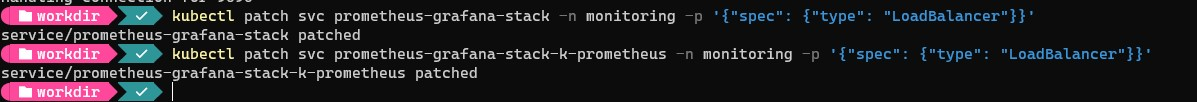
kubectl get svc -n monitoring

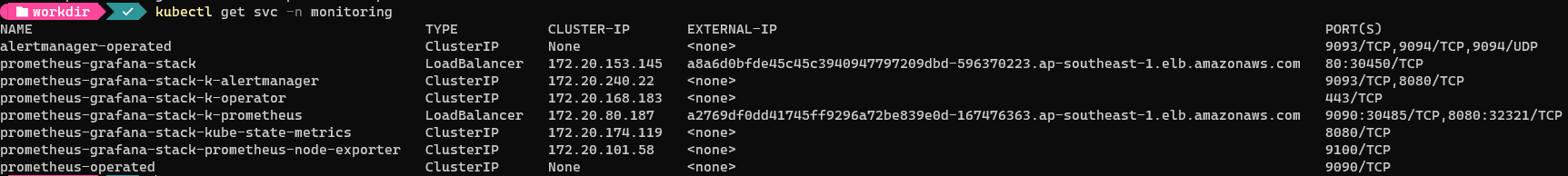


1. Expose prometheus and grafana

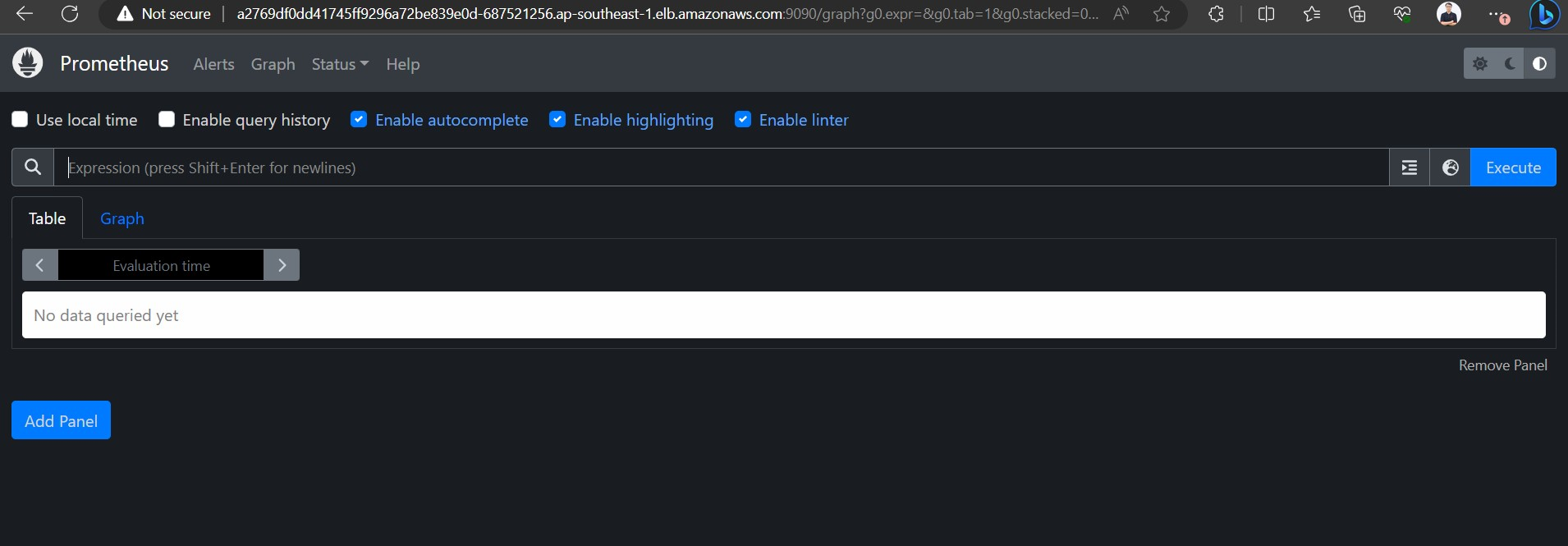
kubectl patch svc prometheus-grafana-stack -n monitoring -p '{"spec": {"type": "LoadBalancer"}}'

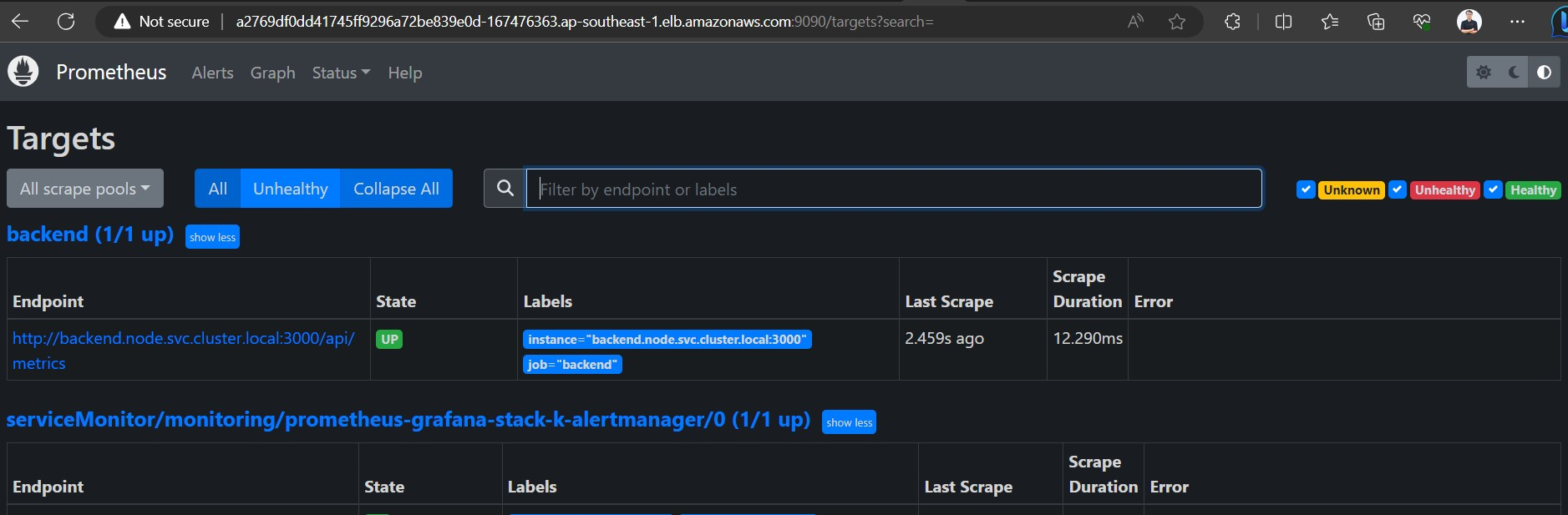
kubectl patch svc prometheus-grafana-stack-k-prometheus -n monitoring -p '{"spec": {"type": "LoadBalancer"}}'

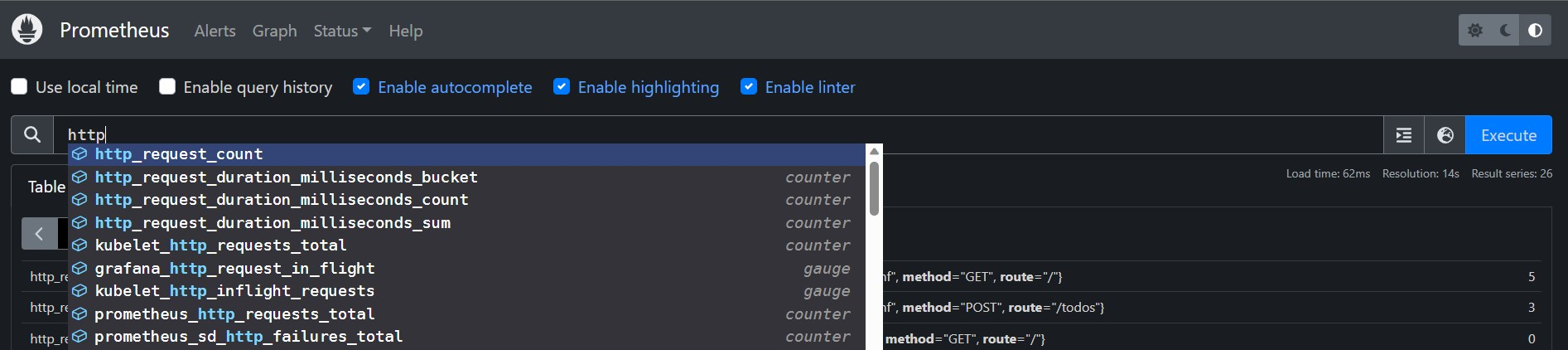




1. Access Prometheus and confirm backend metrics is added





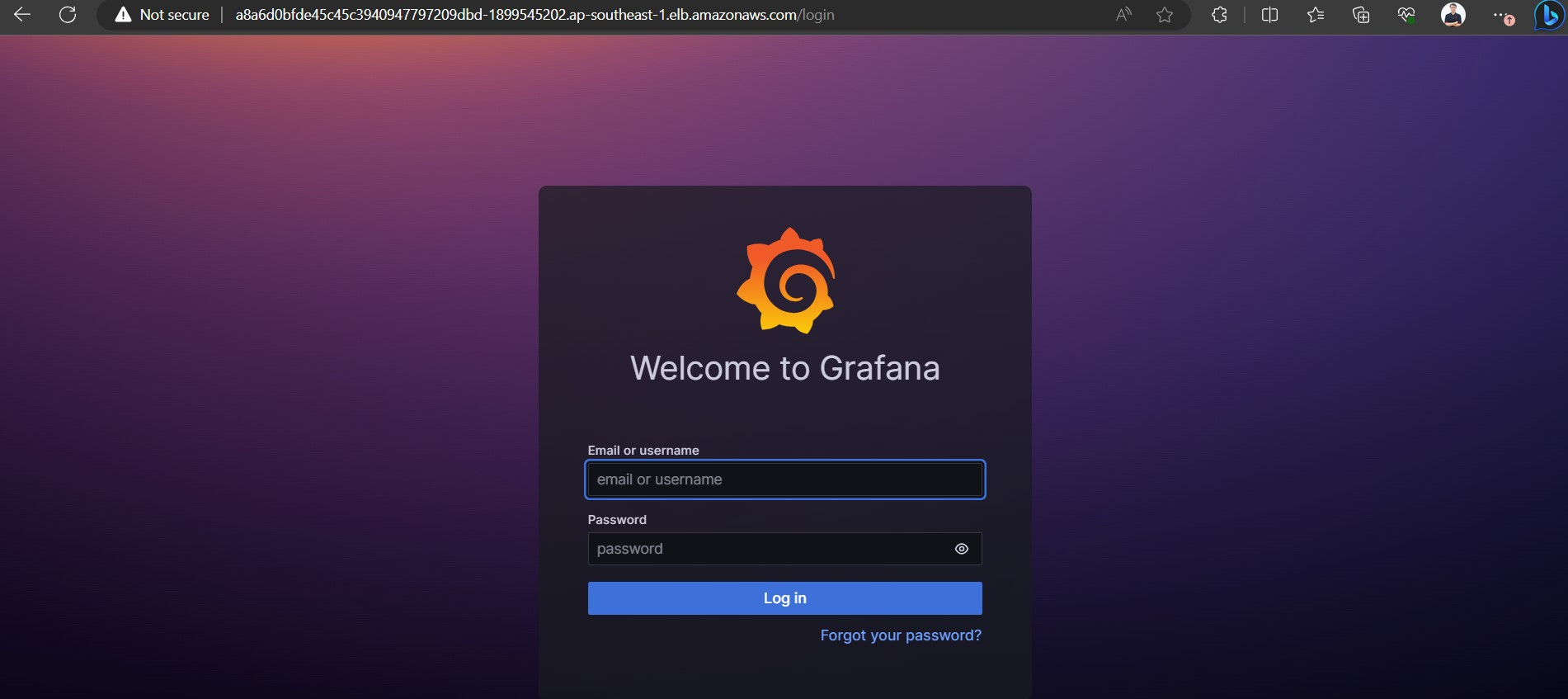


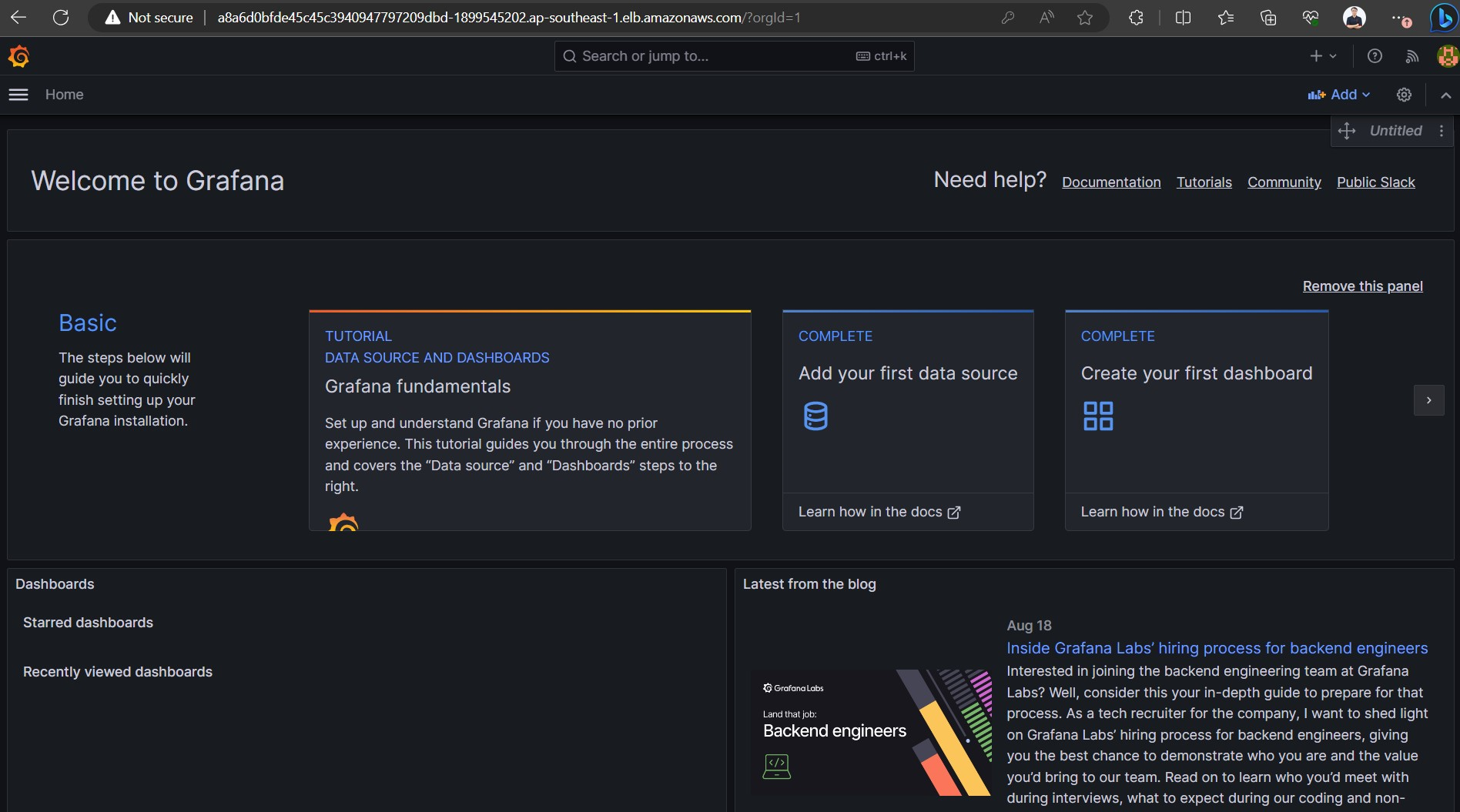
1. Access grafana

Get grafana password:

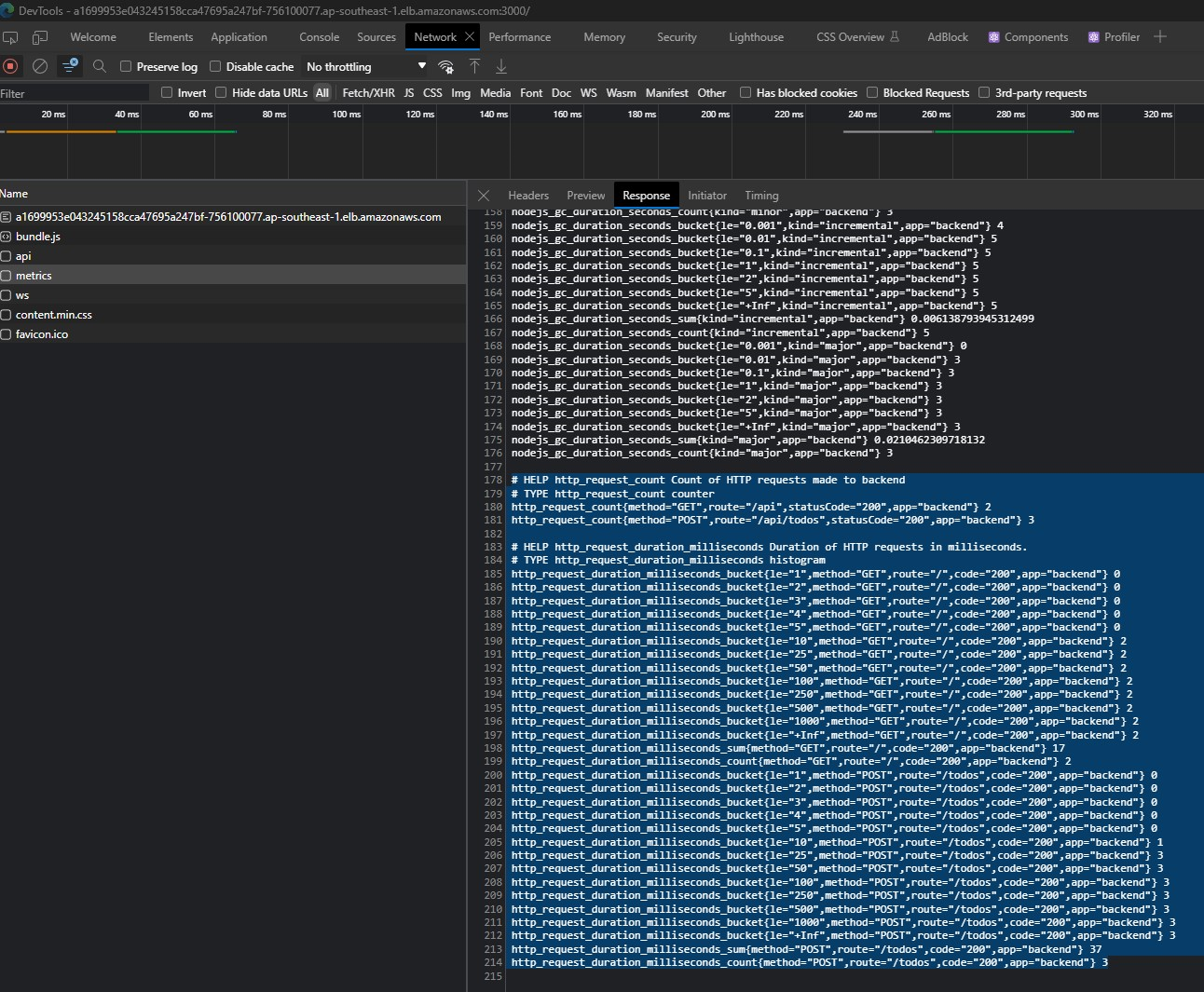
kubectl get secret --namespace monitoring prometheus-grafana-stack -o jsonpath="{.data.admin-password}"

decoded base64: prom-operator





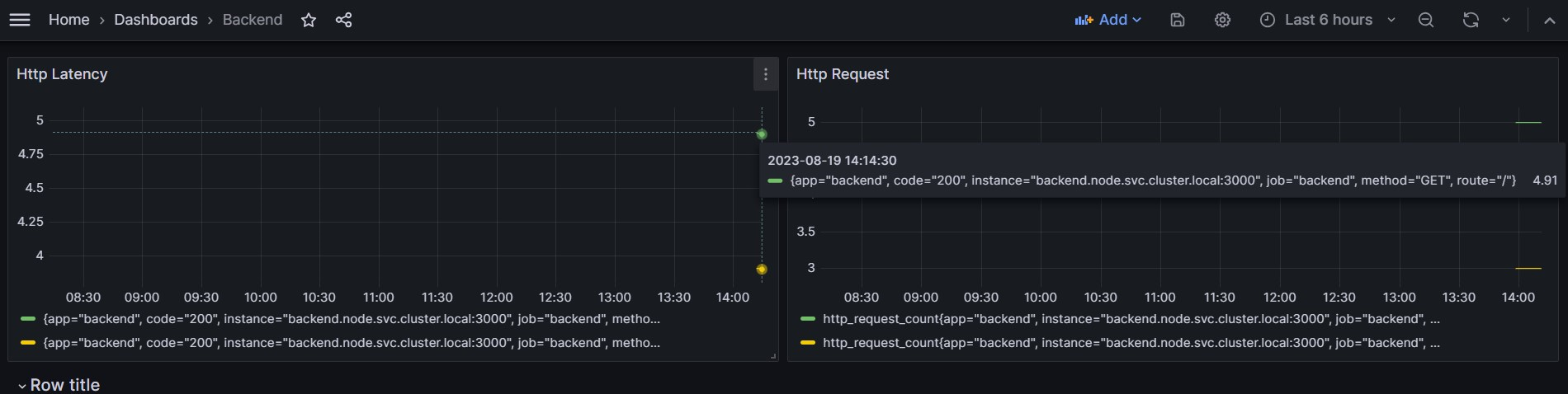
1. Back to app and do something to get some metrics added.

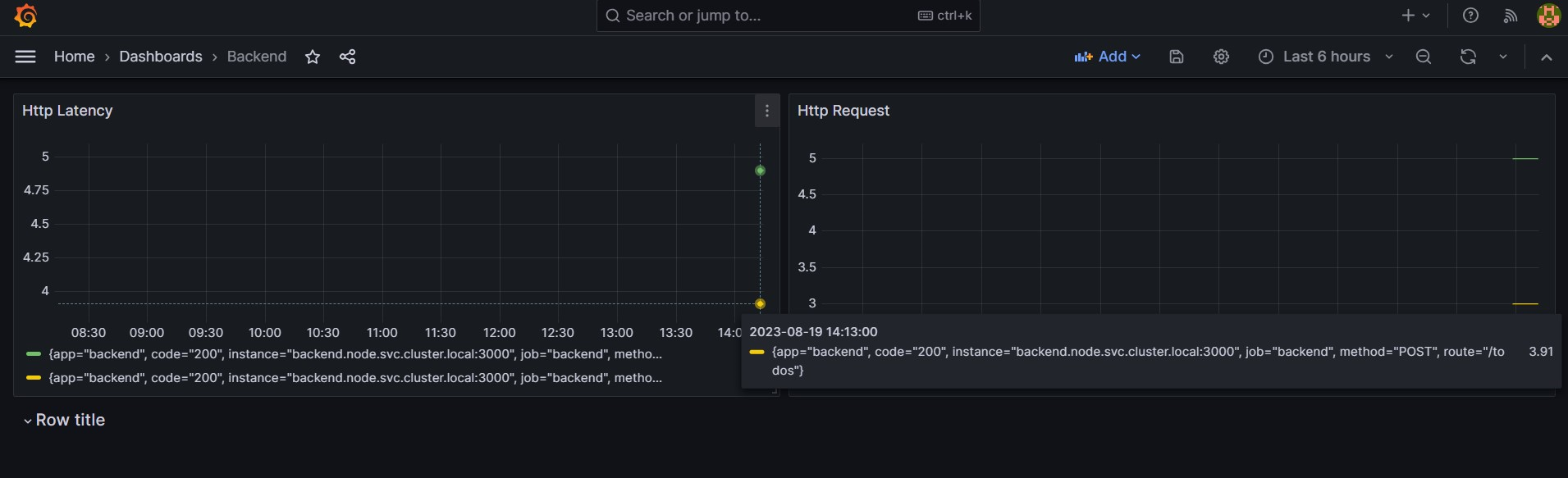


1. Config metrics to a dashboard

Request latency 90% percentage. It will provide 90% percentage request latency of each endpoint.

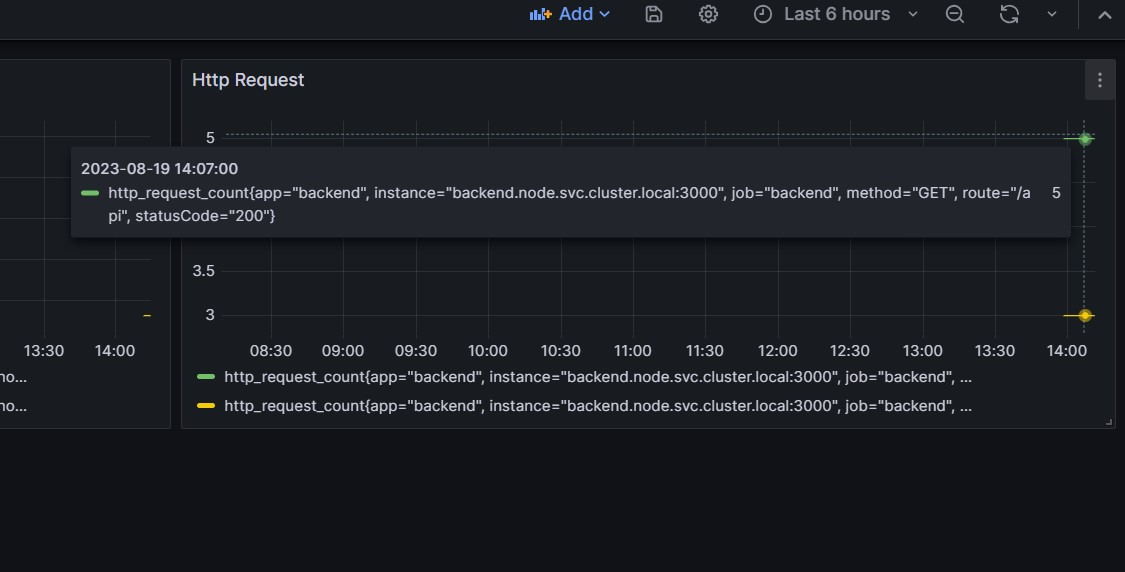
histogram\_quantile(0.9, rate(http\_request\_duration\_milliseconds\_bucket[30m]))





Request count for each method and endpoint

http\_request\_count



---------------