Monitoroing using by Prometheus and Grafana

snap install helm --classic

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

helm repo update

helm install monitoring prometheus-community/kube-prometheus-stack --namespace monitoring2 --create-namespace

kubectl --namespace monitoring2 get pods -l "release=monitoring"

kubectl get pods -n monitoring2

kubectl get secret -n monitoring2 monitoring-grafana -o jsonpath="{.data.admin-password}" | base64 --decode ; echo

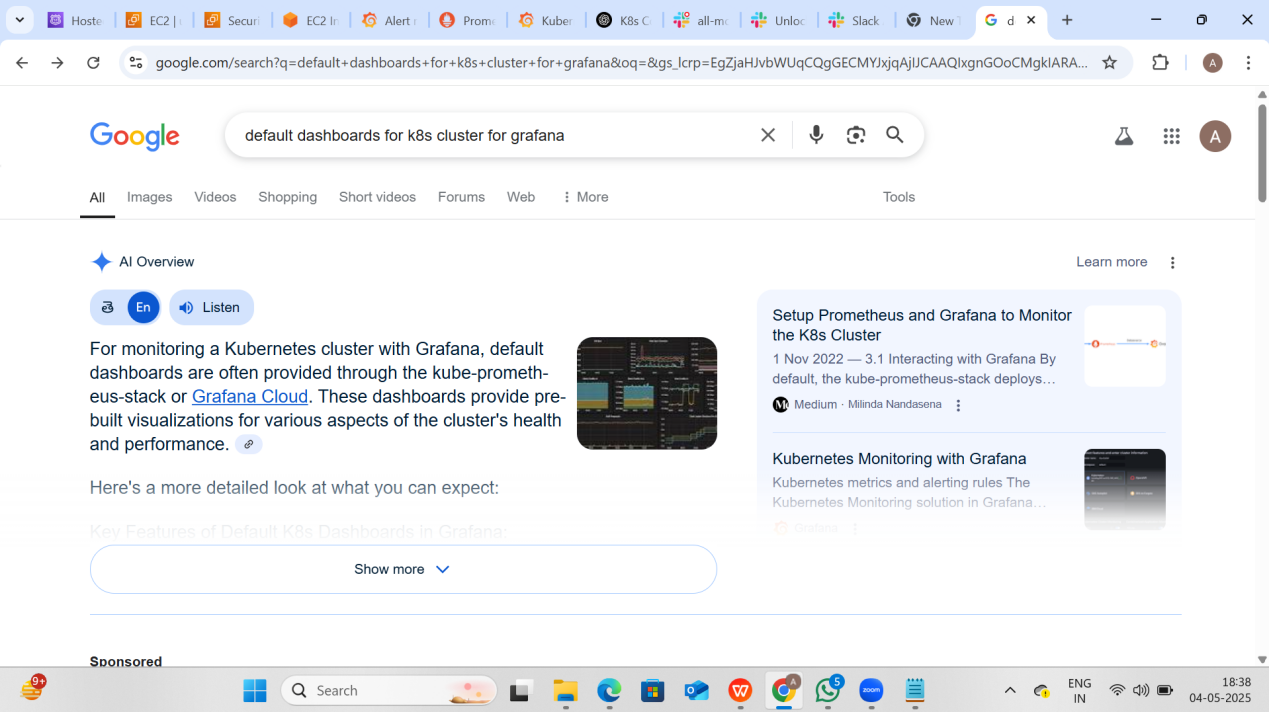
kubectl edit svc -n monitoring2 monitoring-grafana

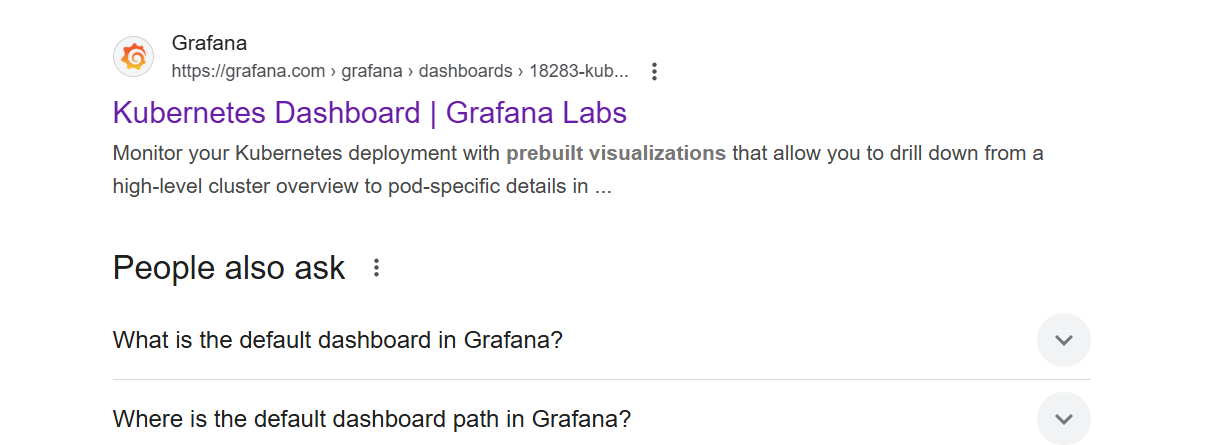
kubectl get svc -n monitoring2 monitoring-grafana

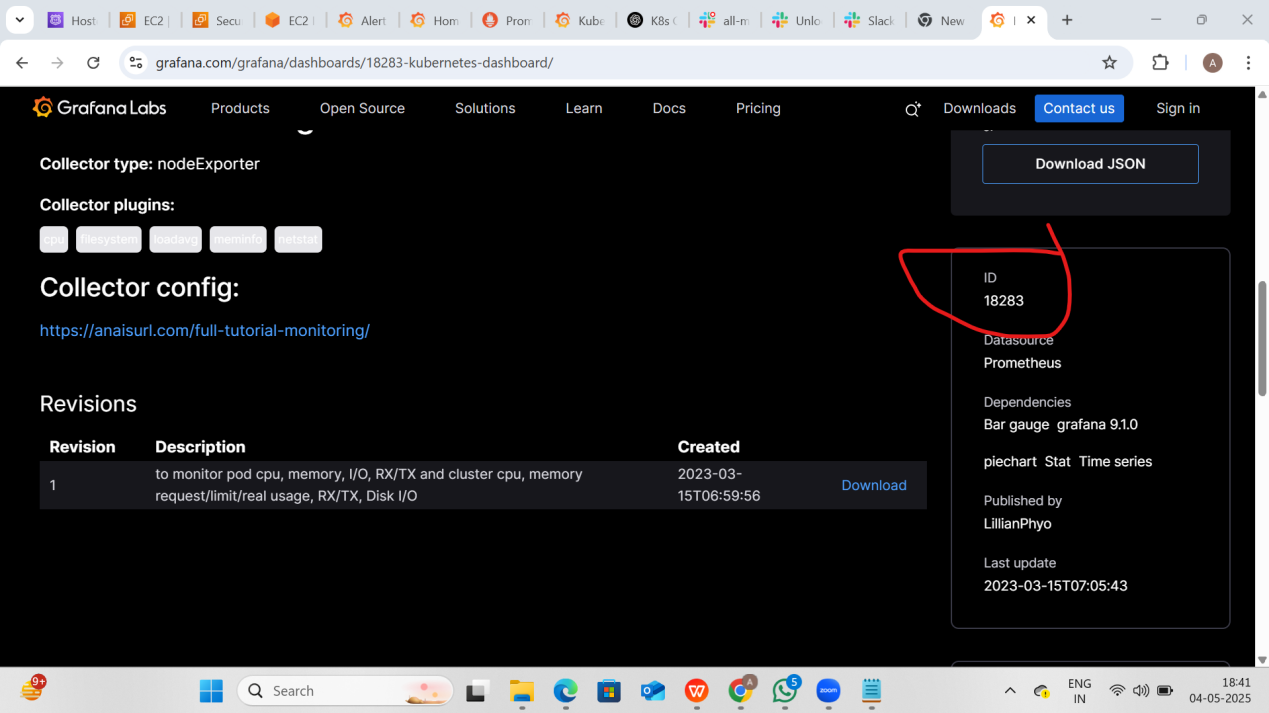
kubectl edit svc -n monitoring2 monitoring-kube-prometheus-prometheus

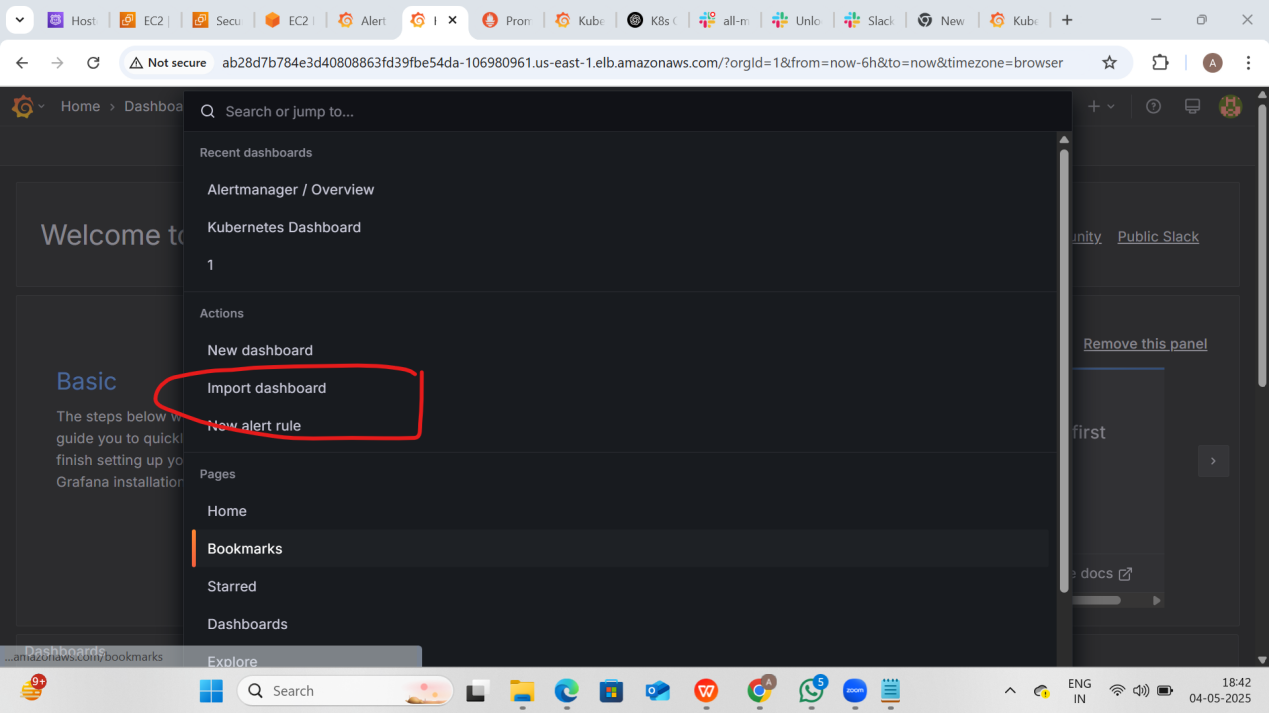
kubectl get svc -n monitoring2 monitoring-kube-prometheus-prometheus

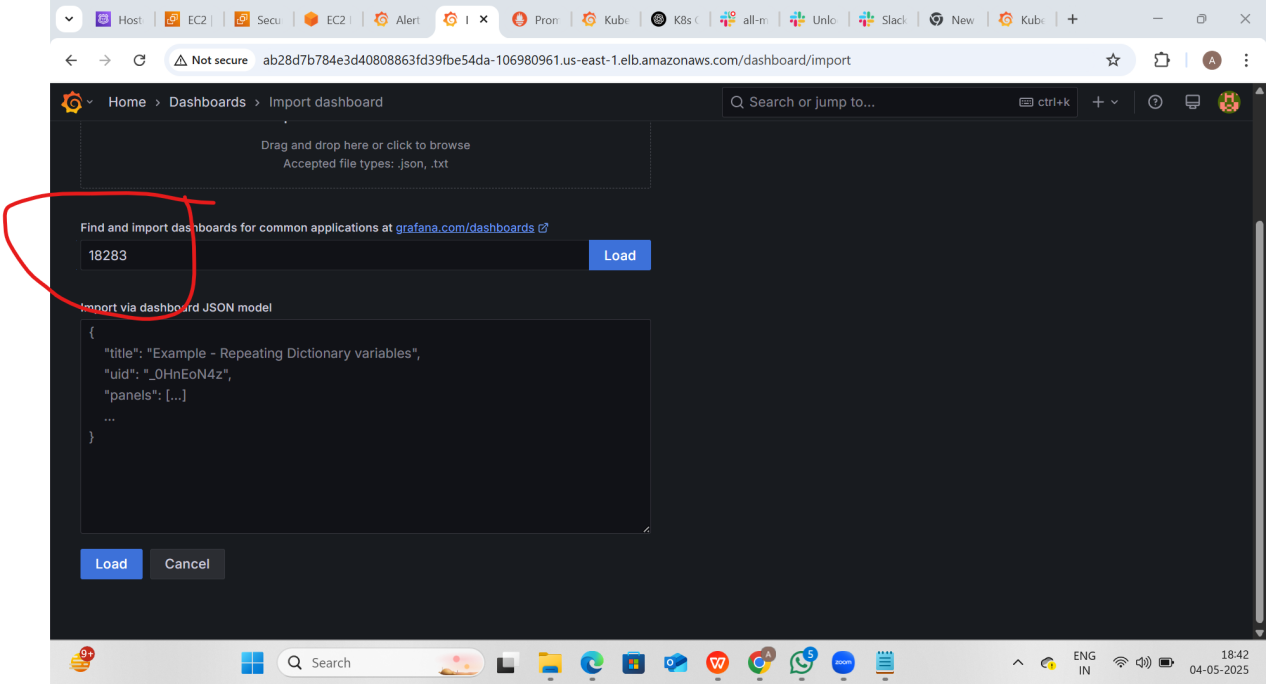
kubectl get pods -n kube-system

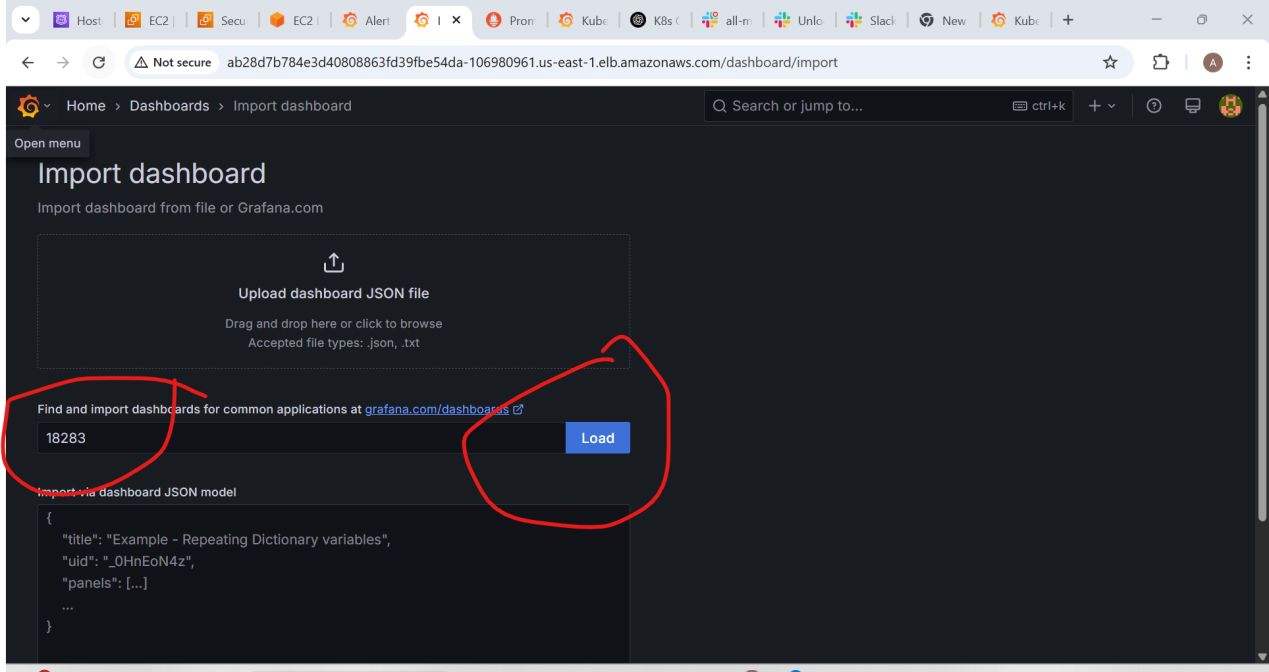




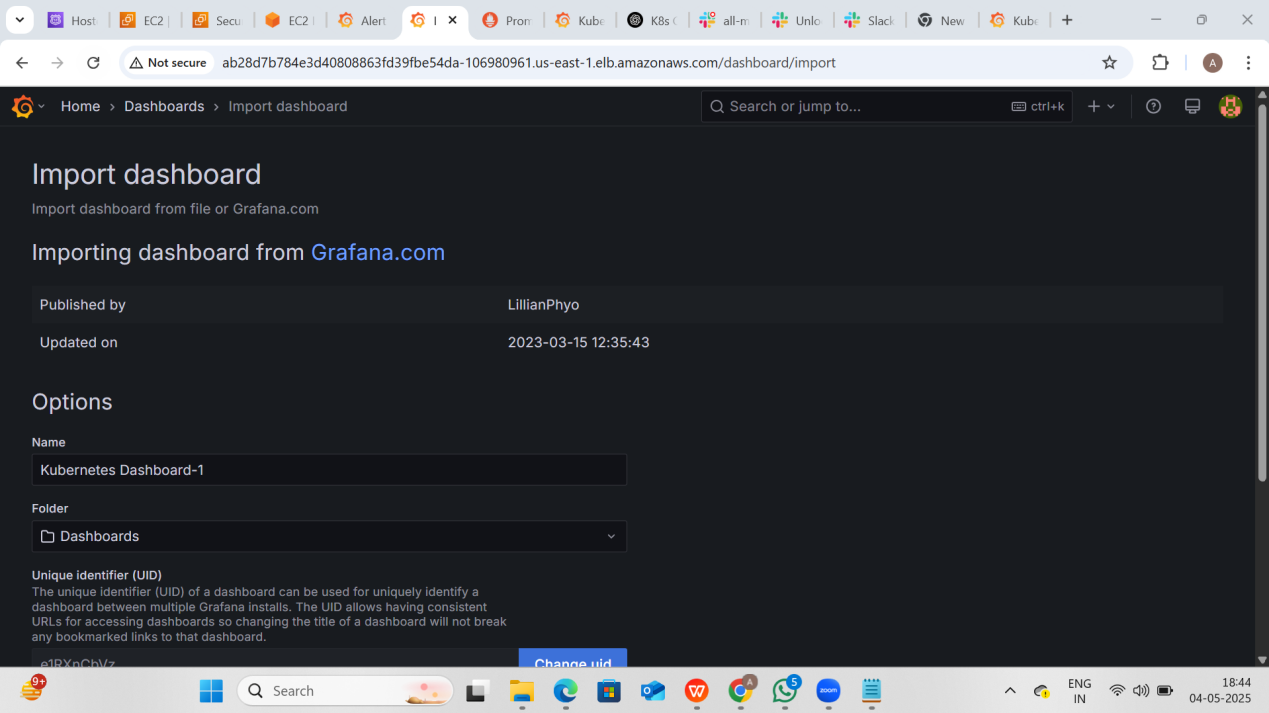


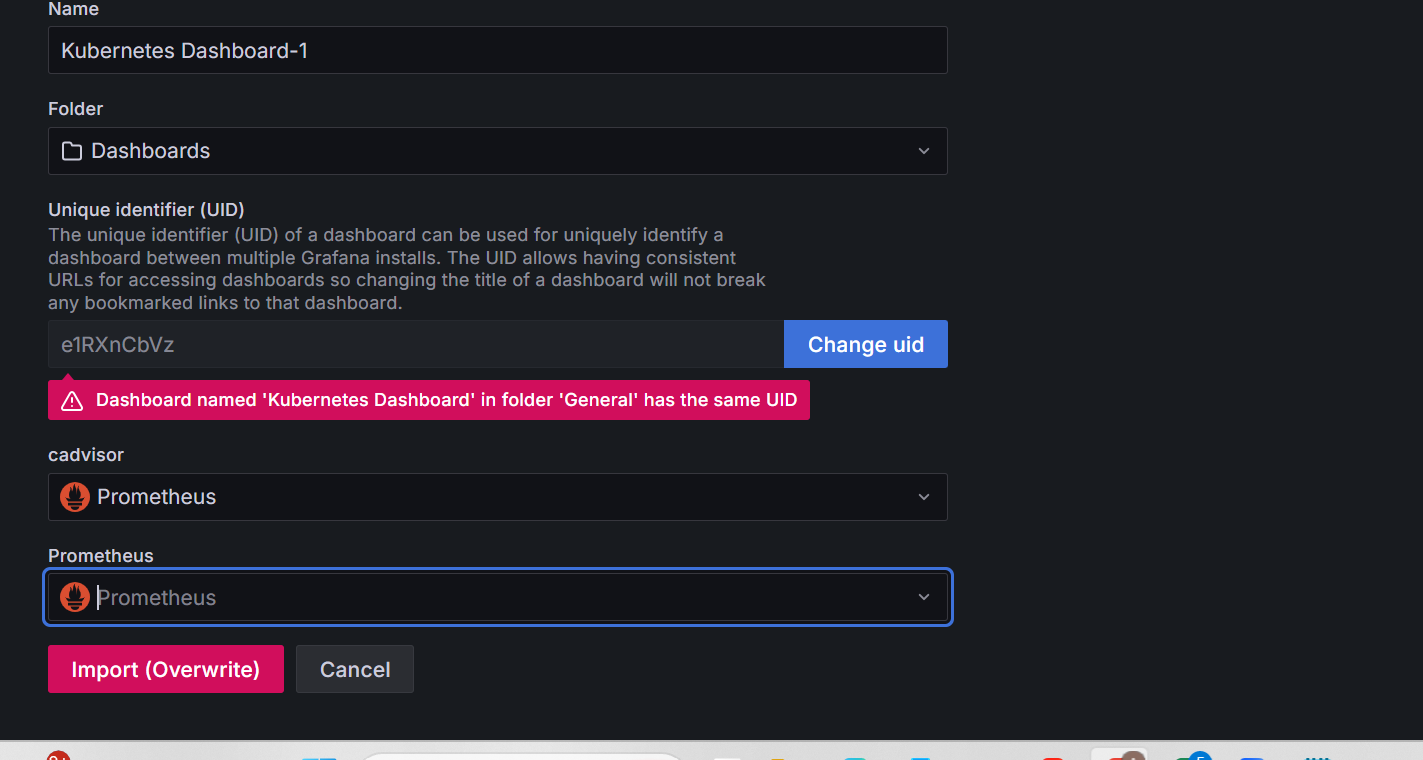


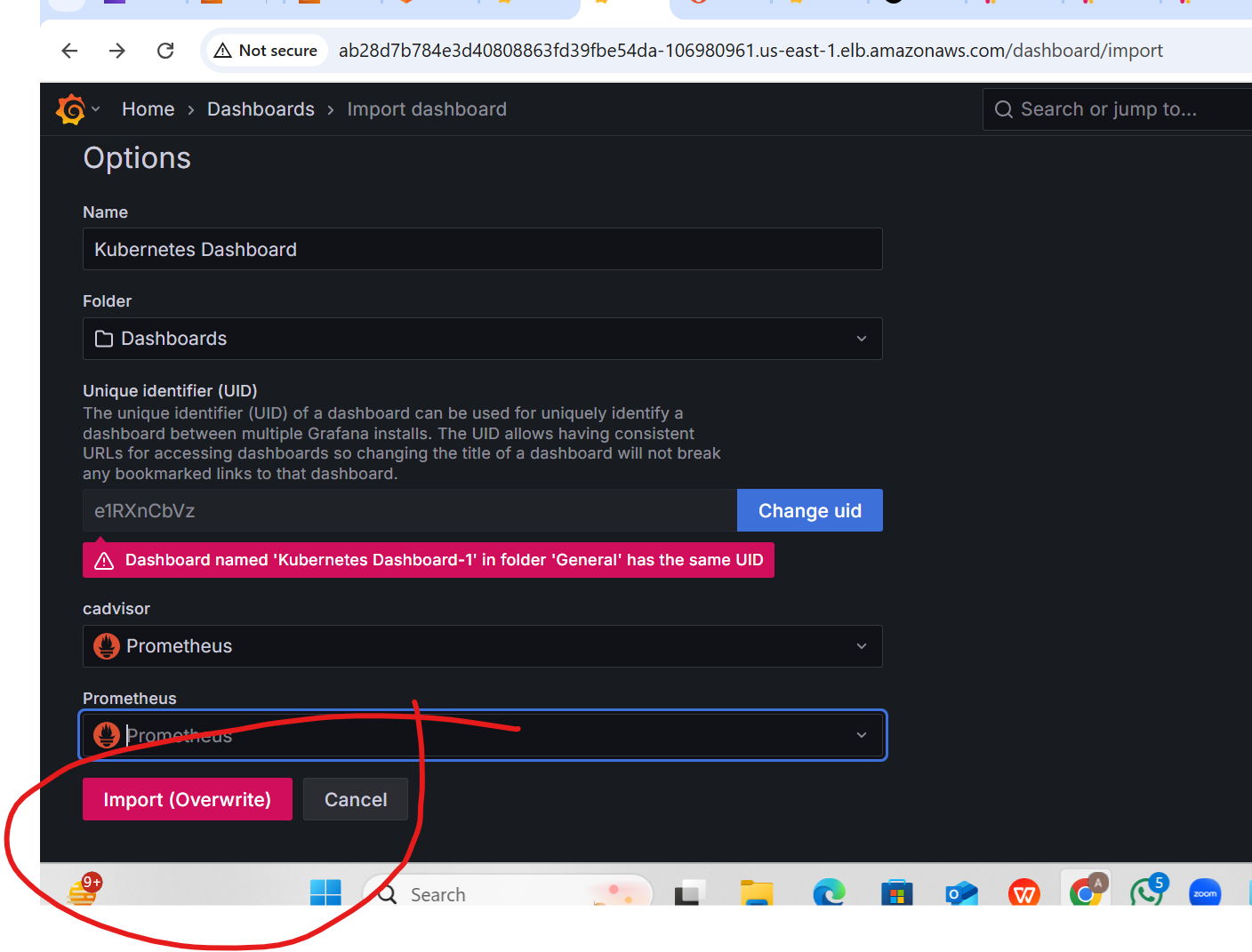




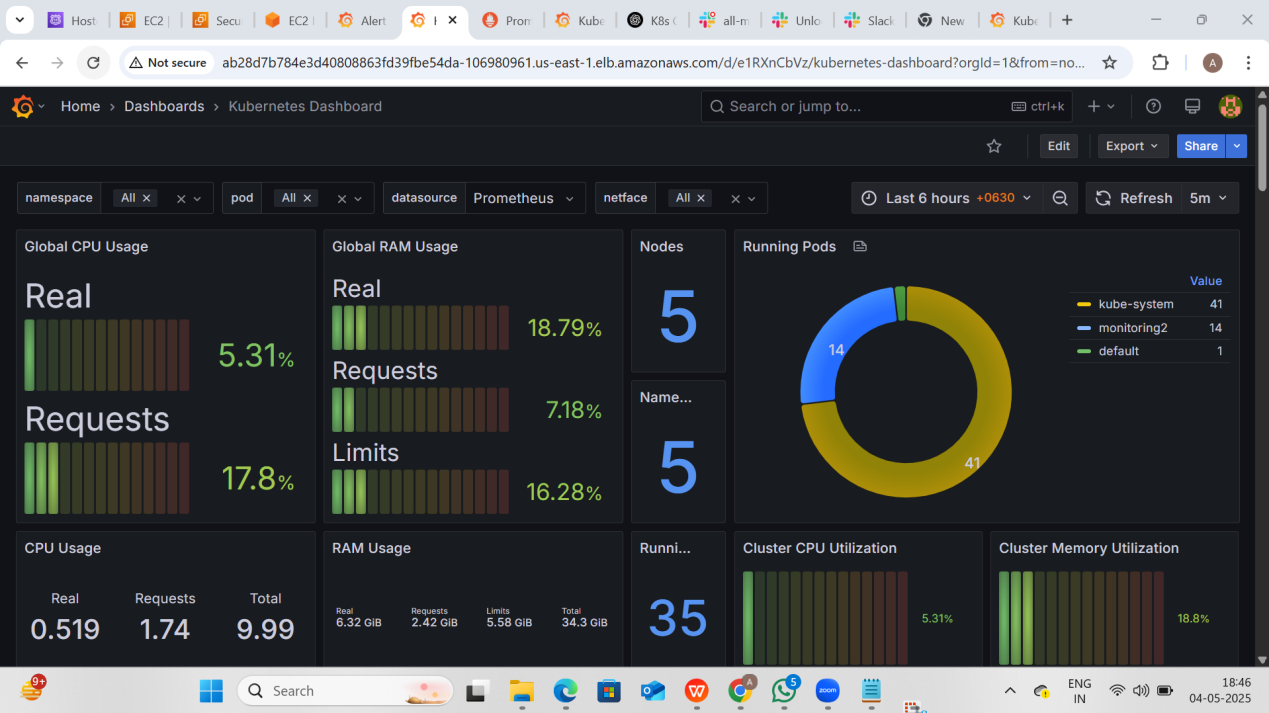
Click on load





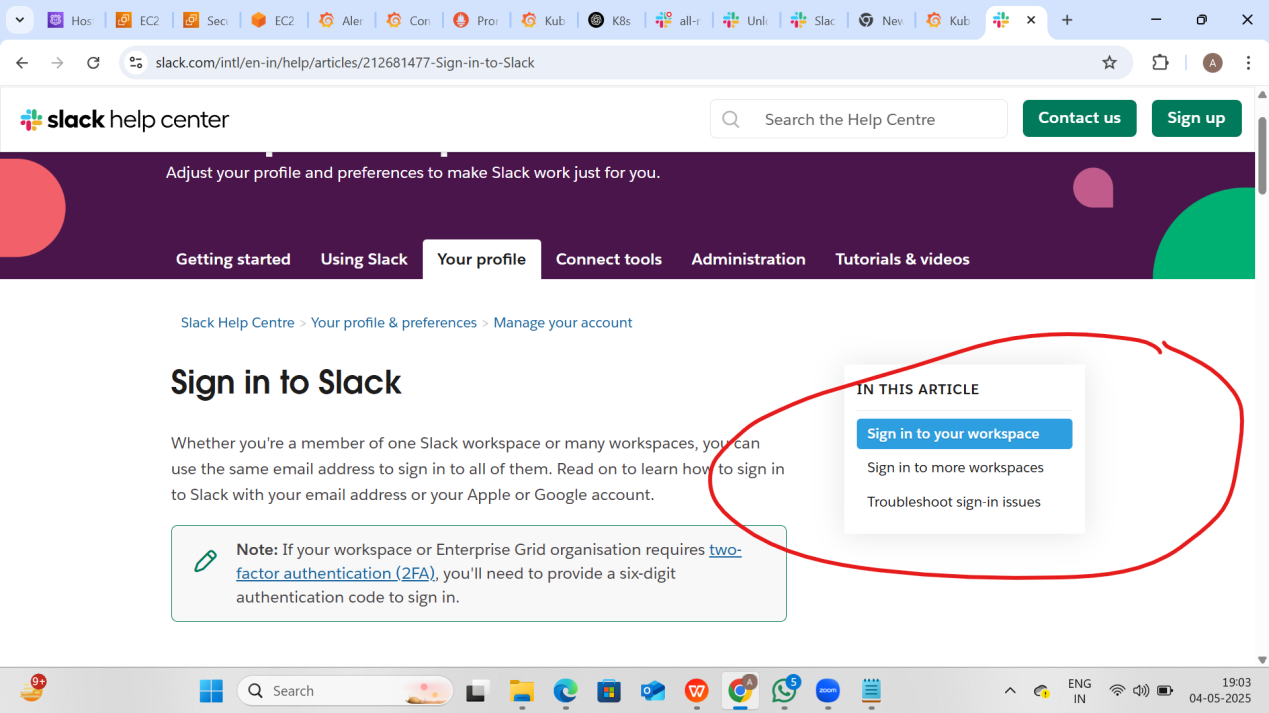


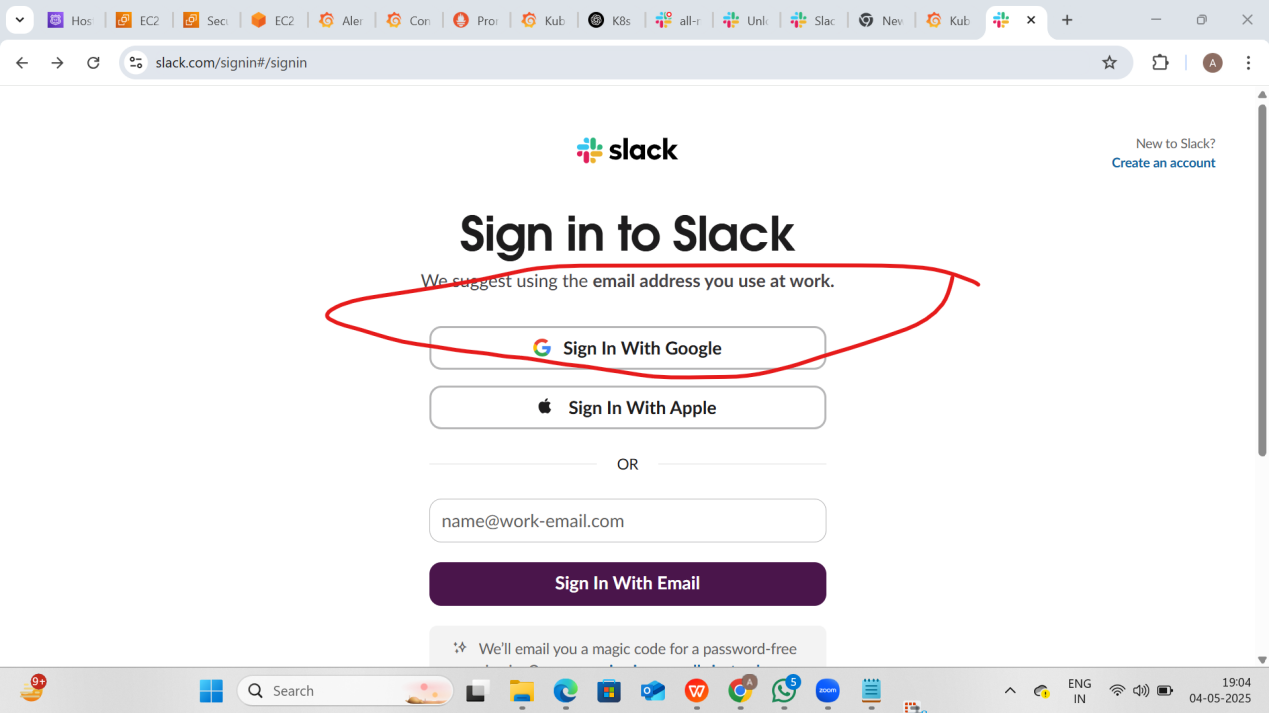
Click on import

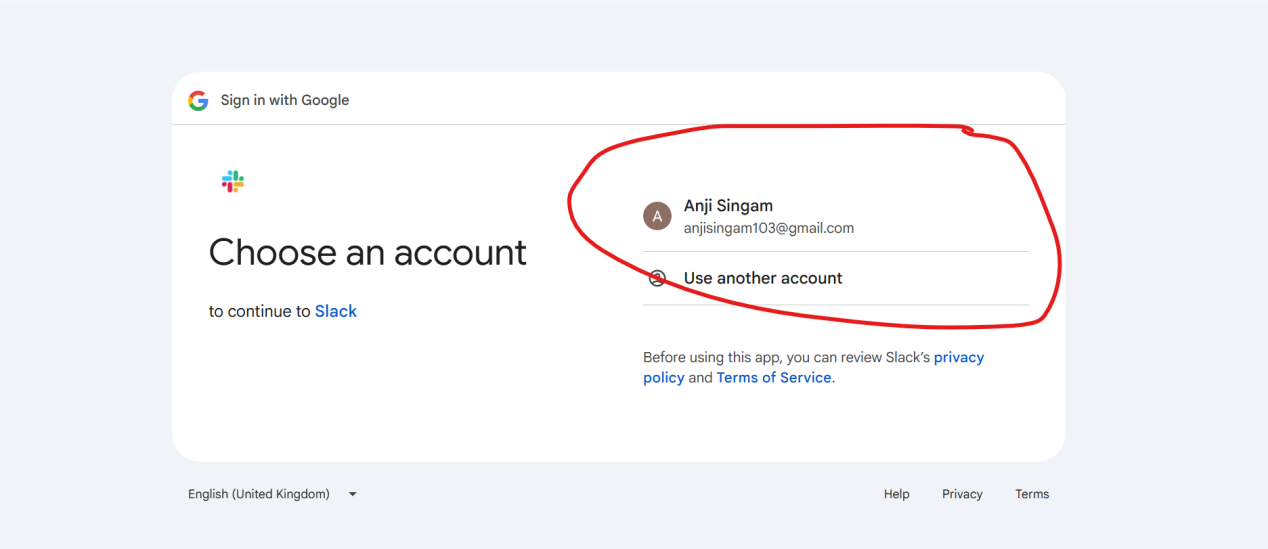


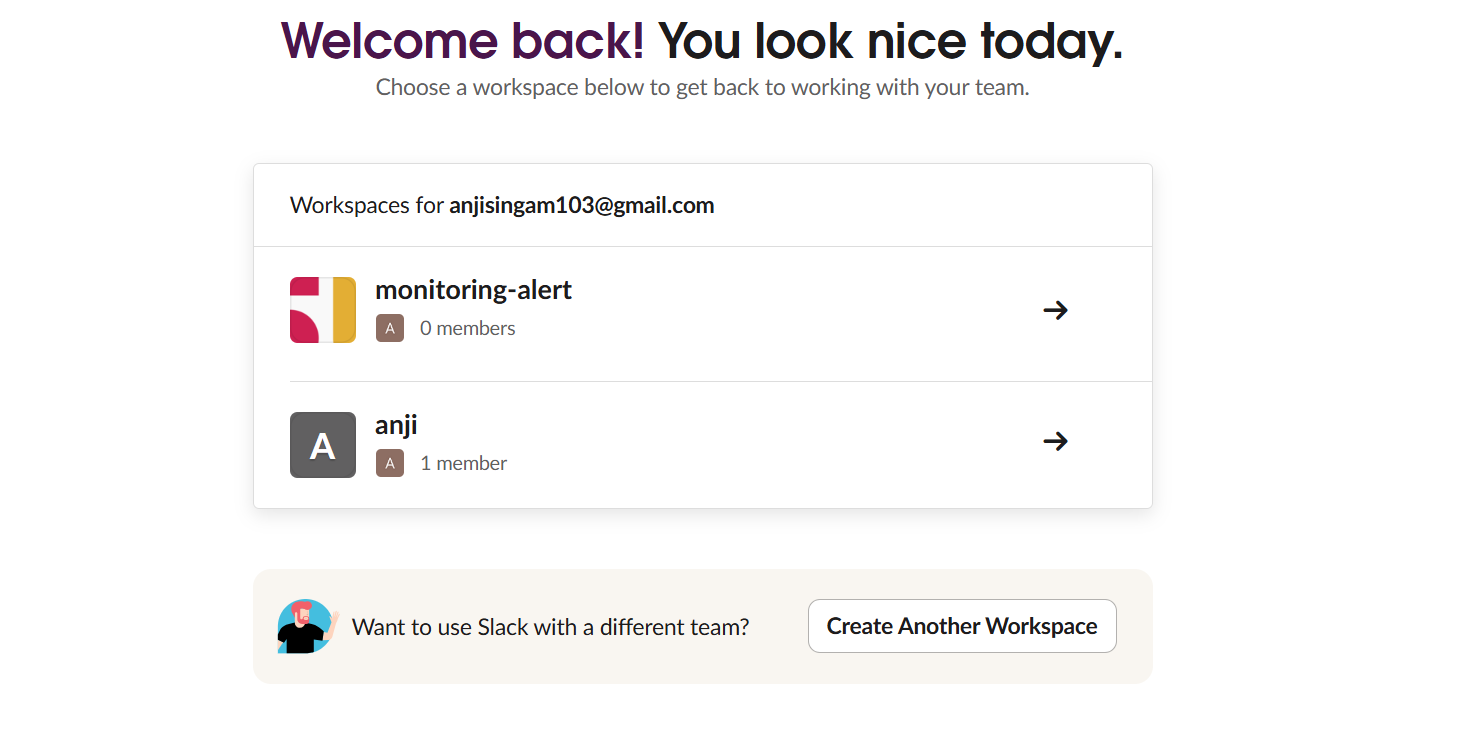
Sign in to slack account

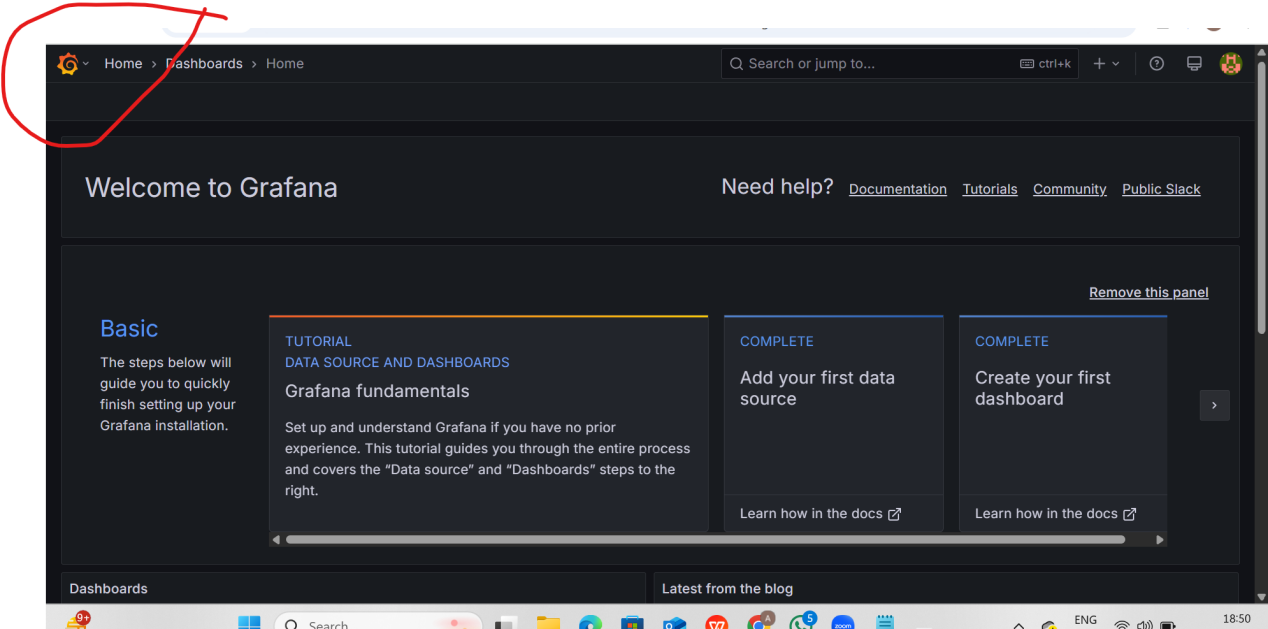




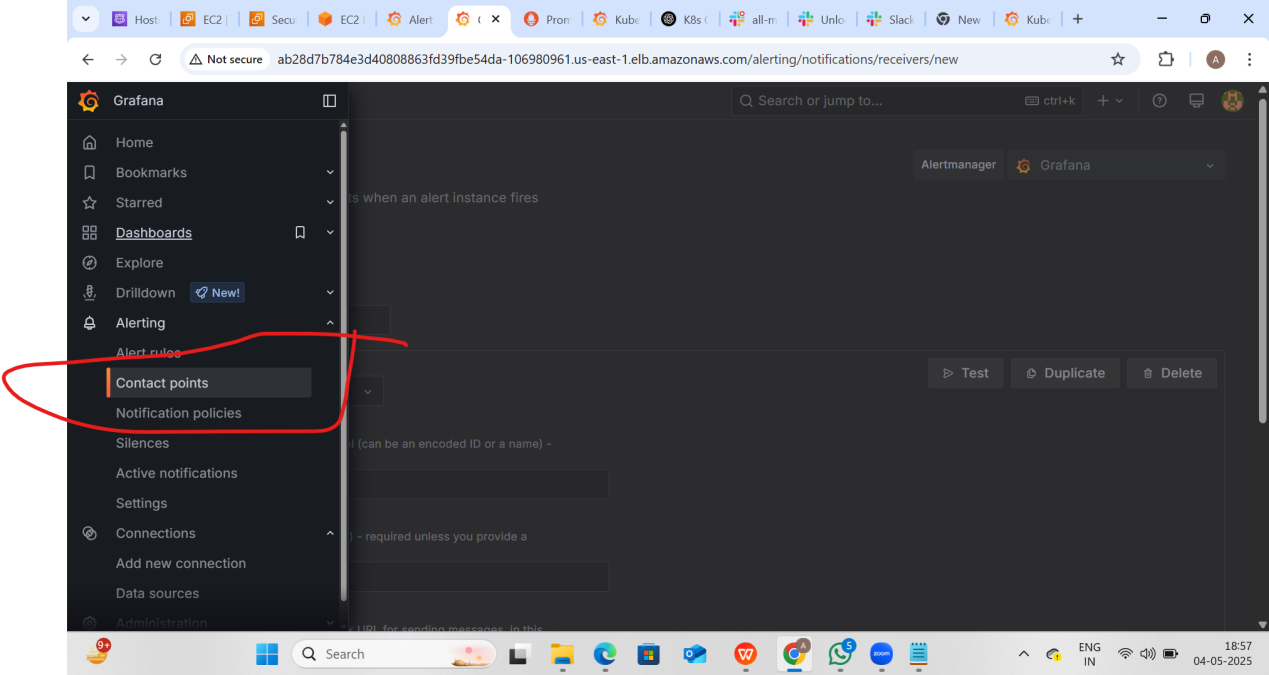




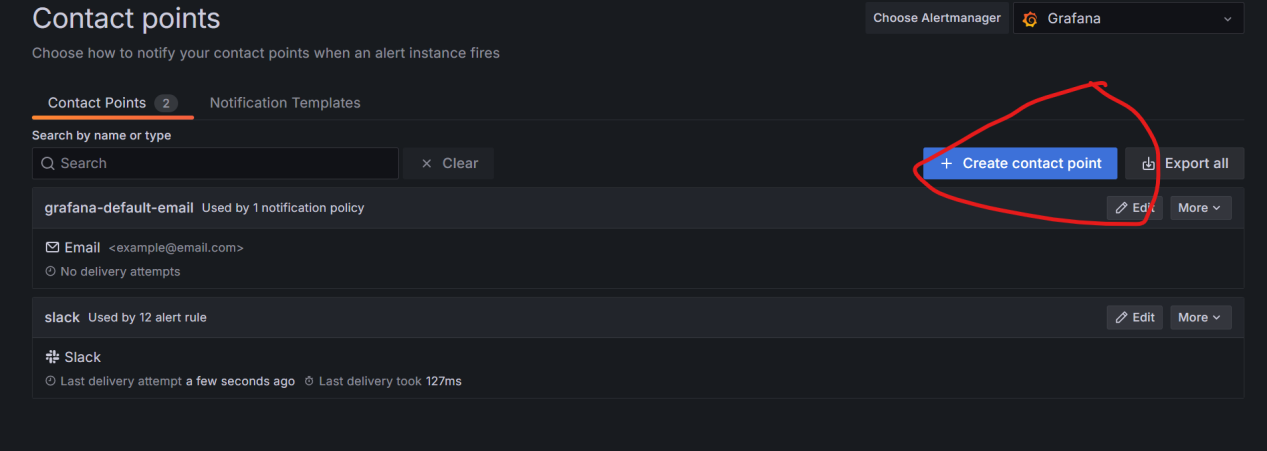


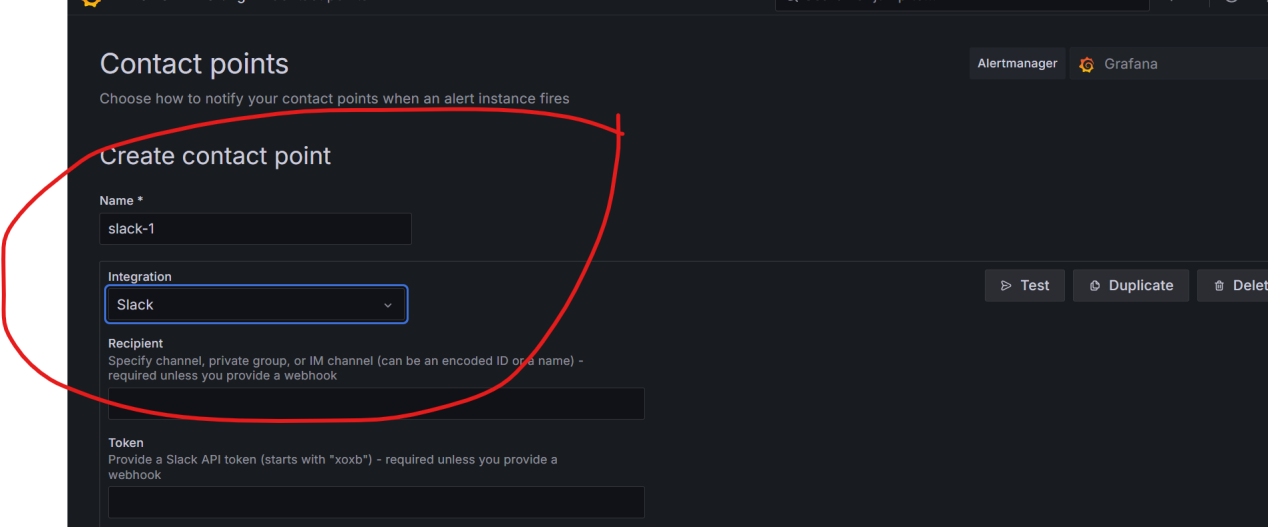


Click on gear icon



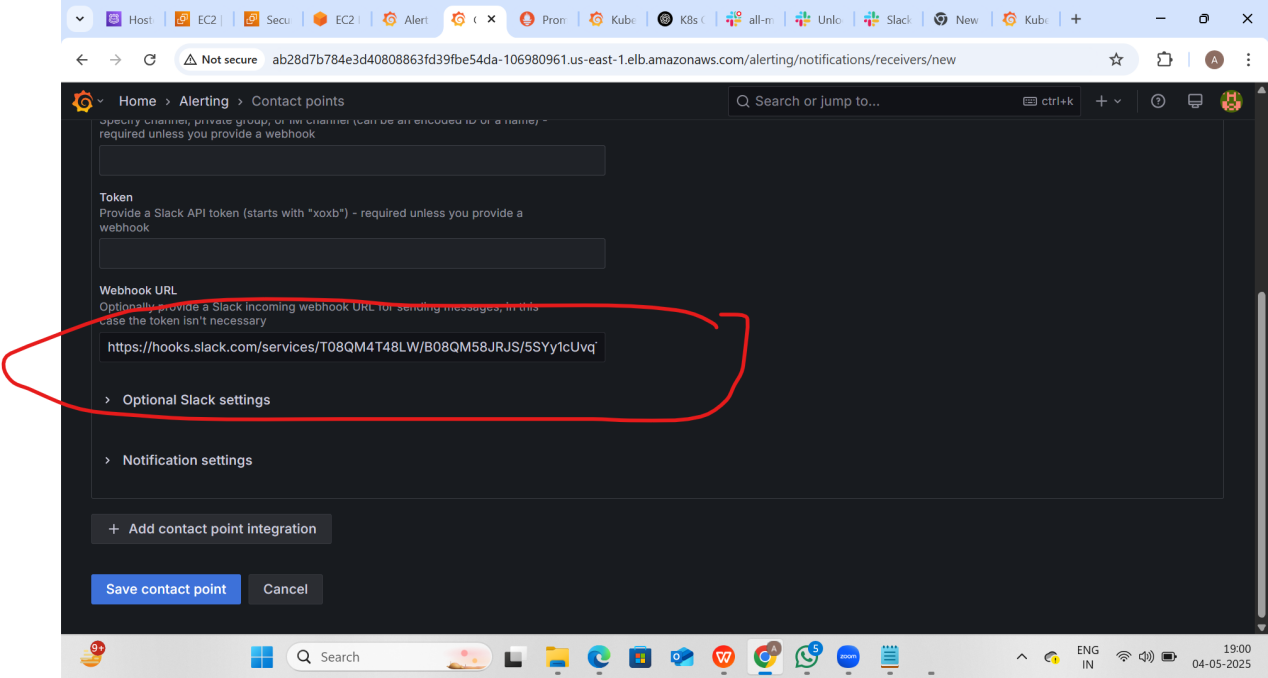
Click on contact points

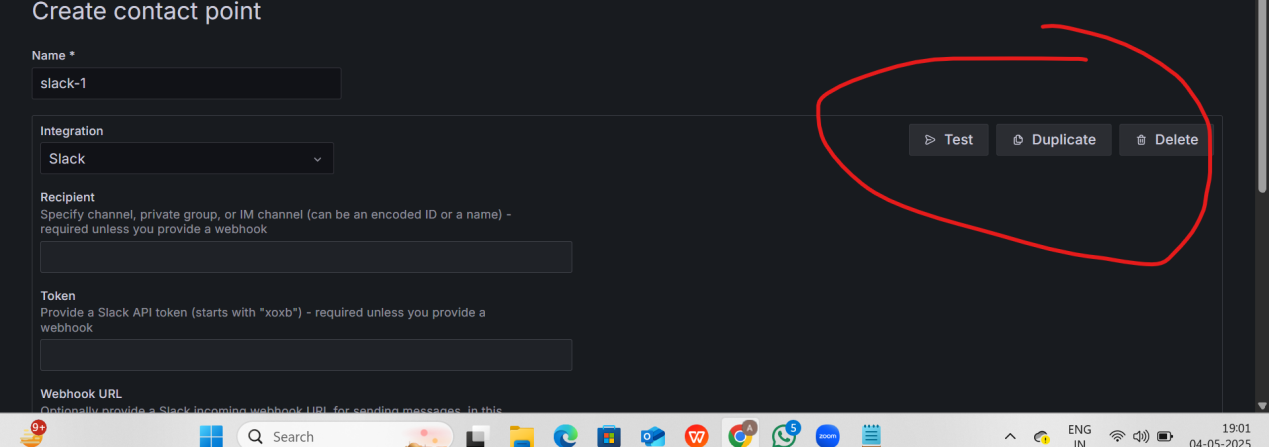




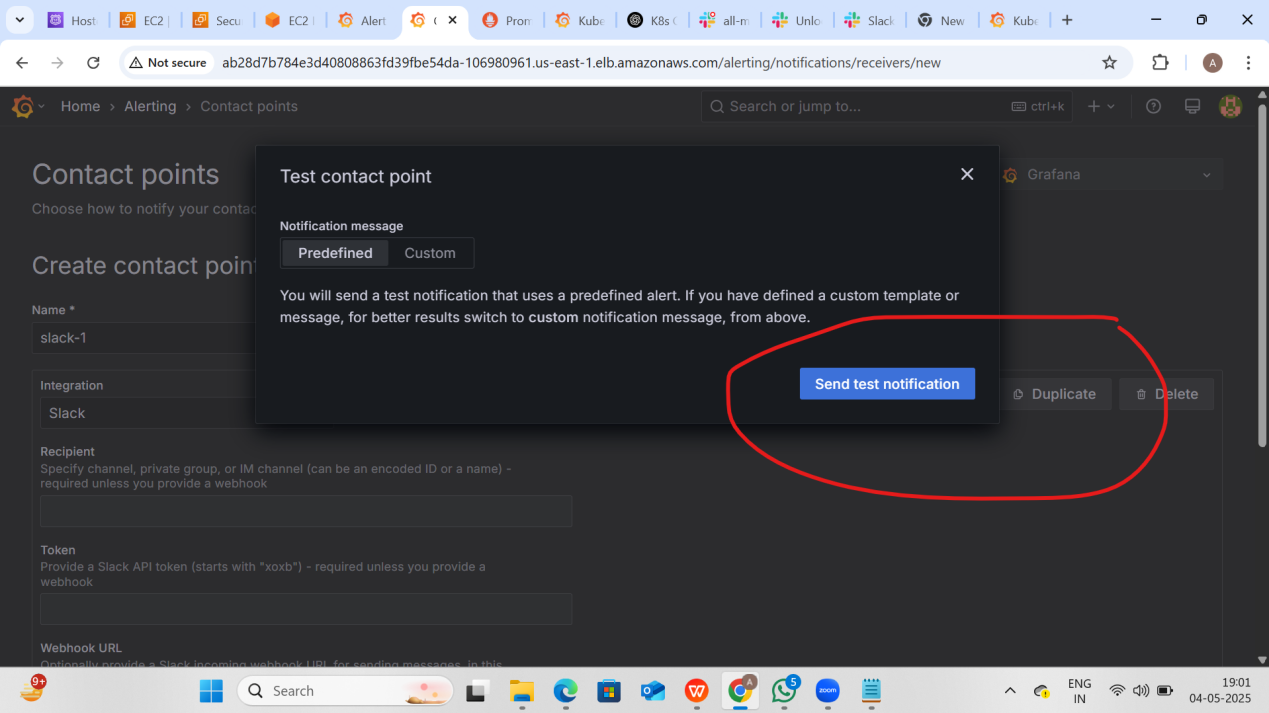
Webhok URL

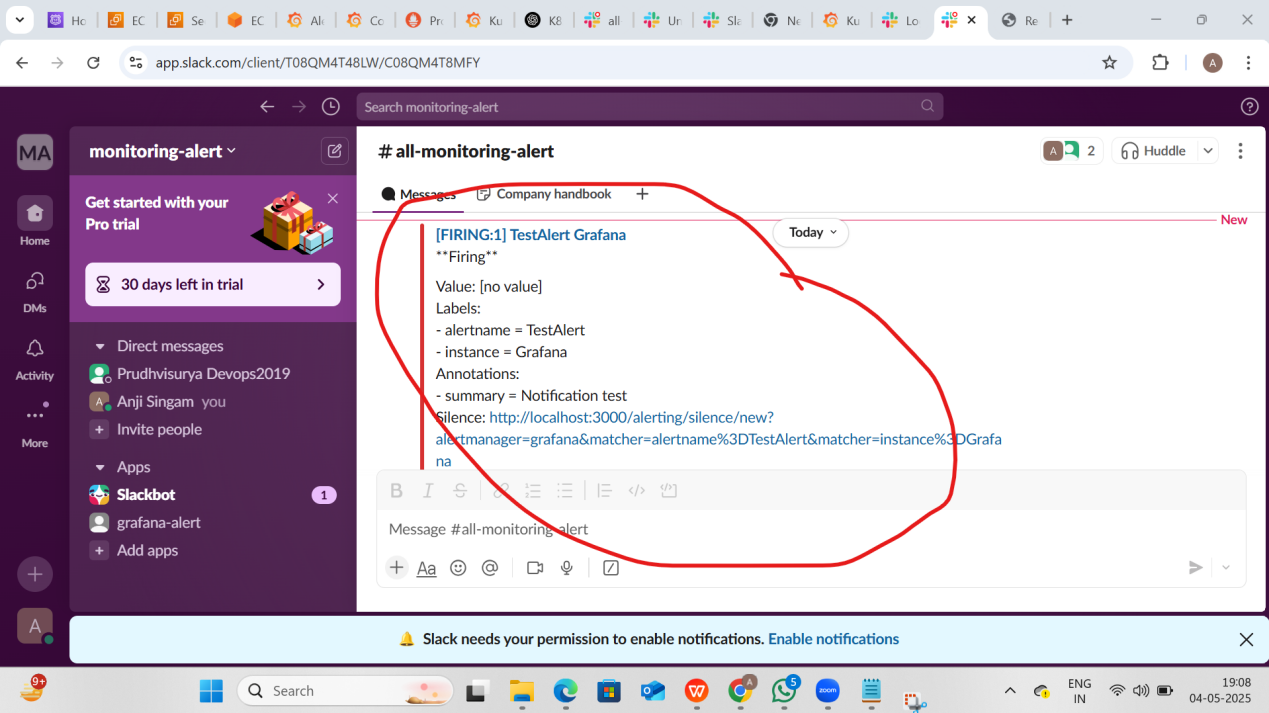
<https://hooks.slack.com/services/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>



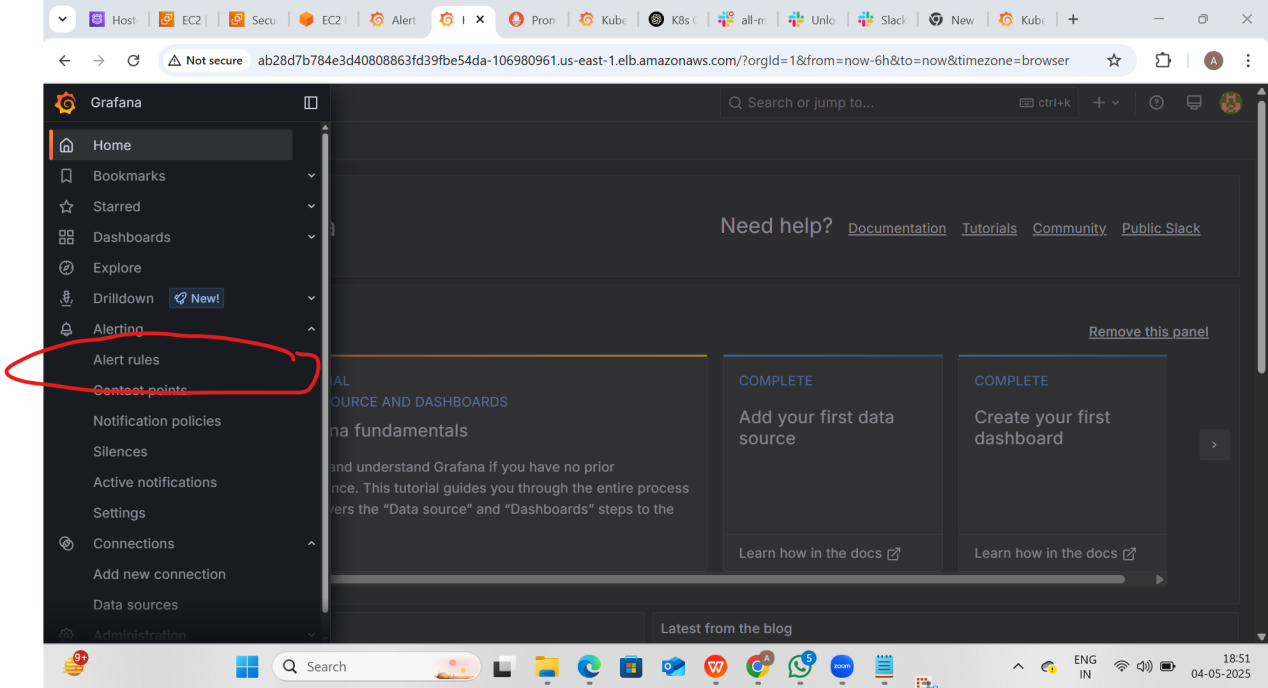


Click on test

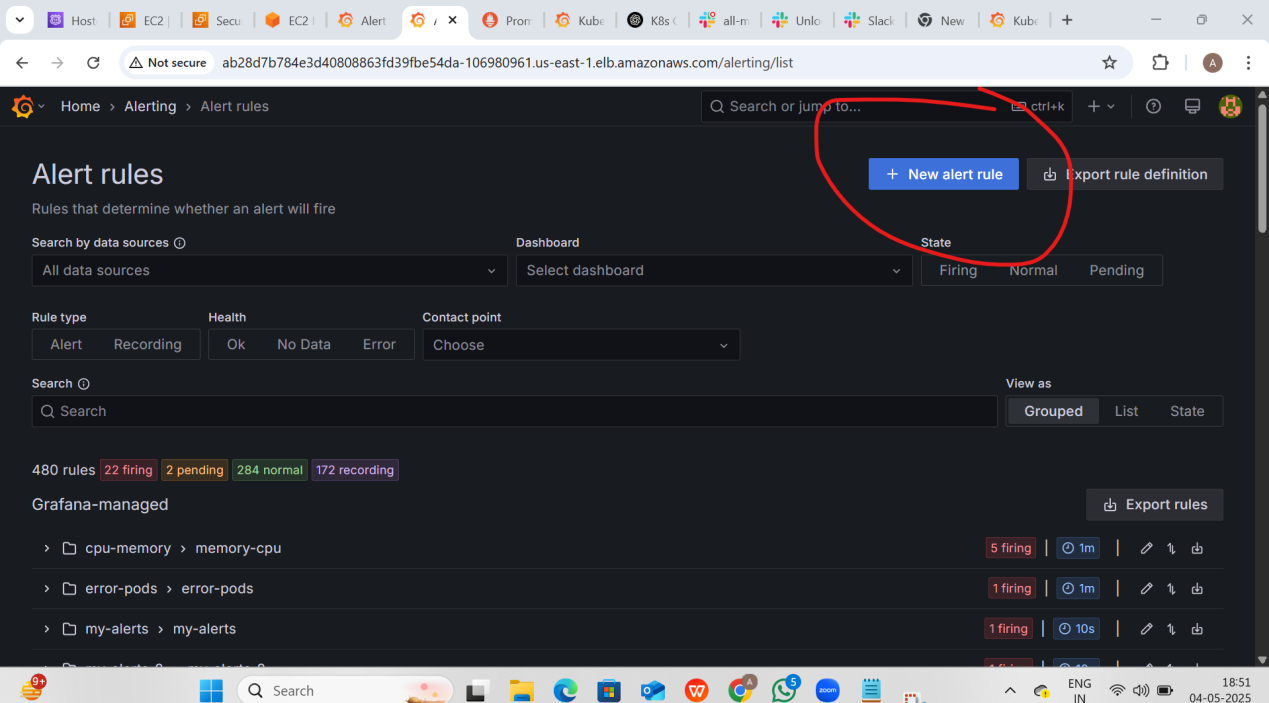




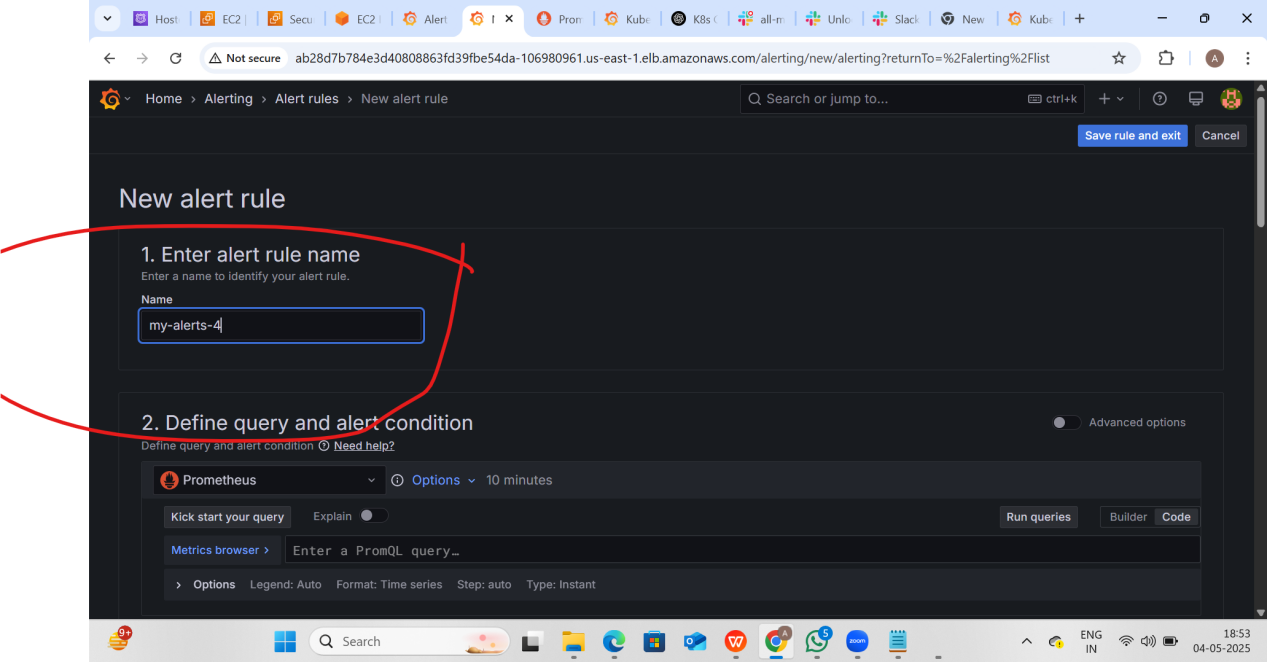
Test alert is received in slack



Click on alert rules



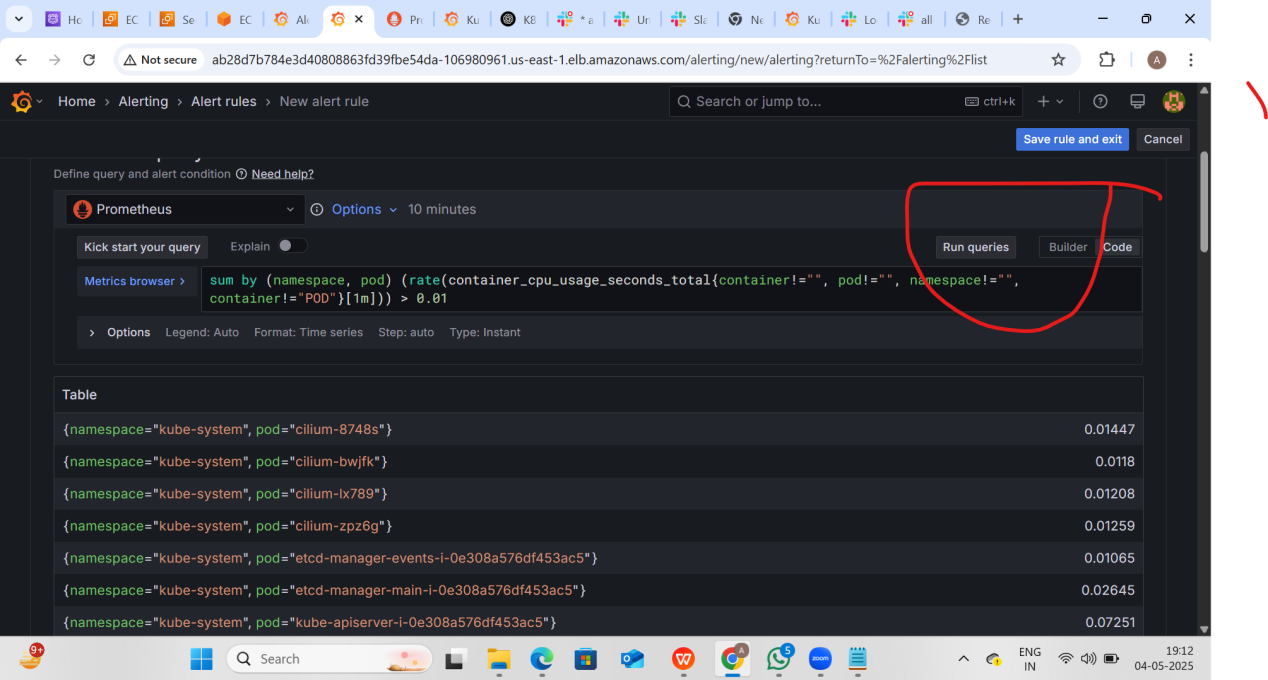
Click on **new alert rule**



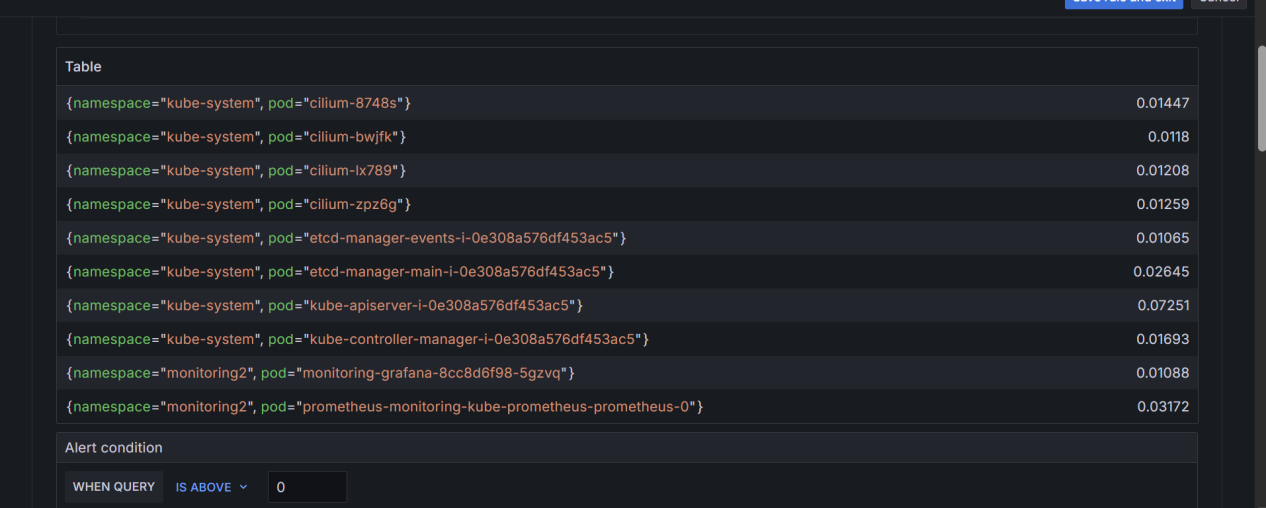
Paste in query below PromQL

sum by (namespace, pod) (rate(container\_cpu\_usage\_seconds\_total{container!="", pod!="", namespace!="", container!="POD"}[1m])) > 0.01

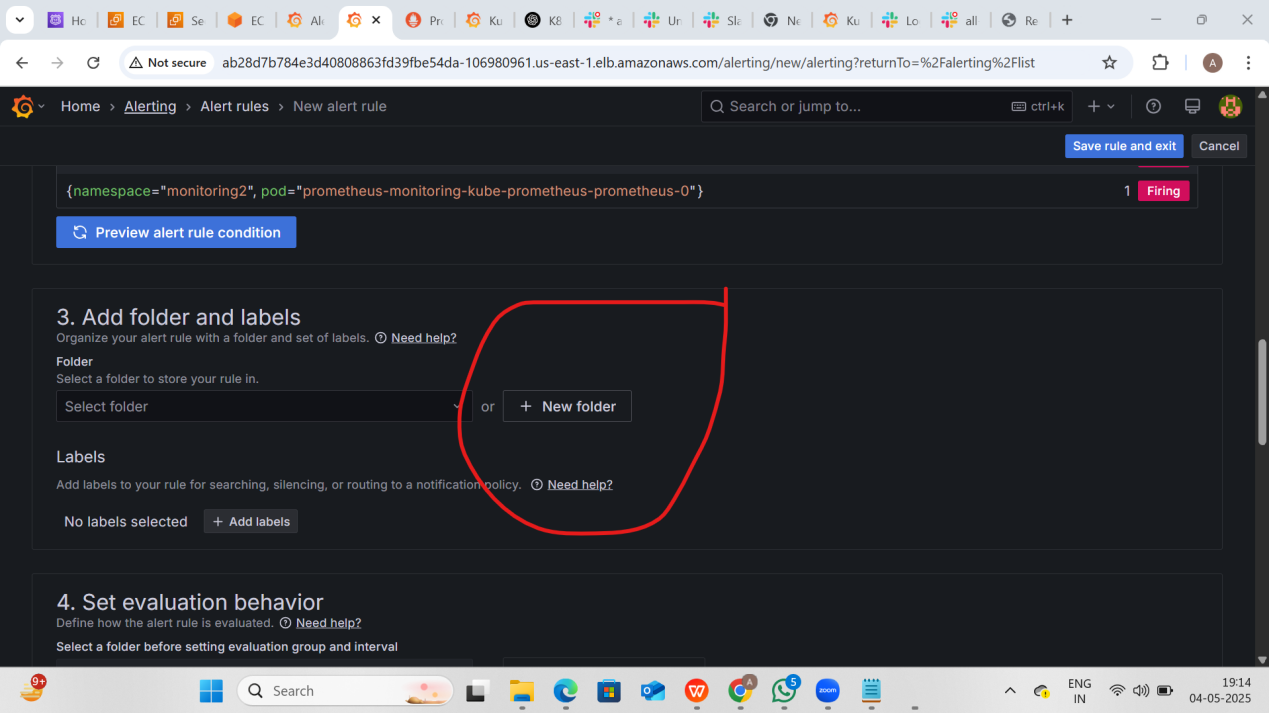




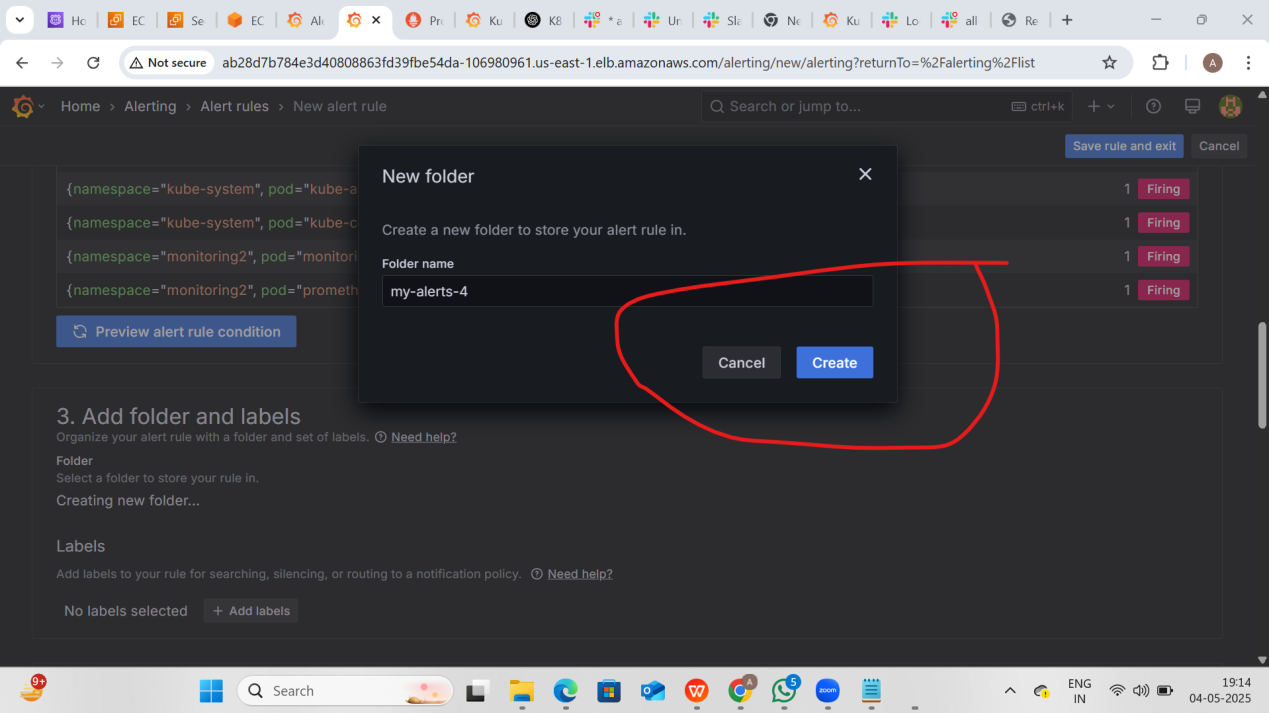
Click on **run queries**





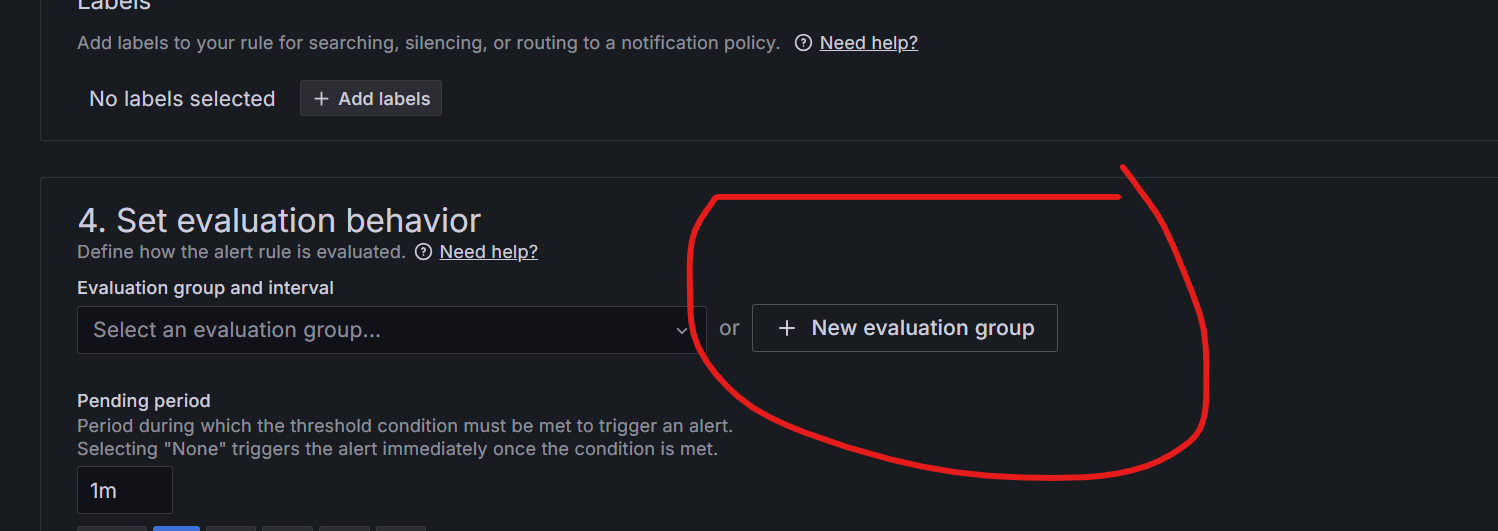


Click on **new floder**

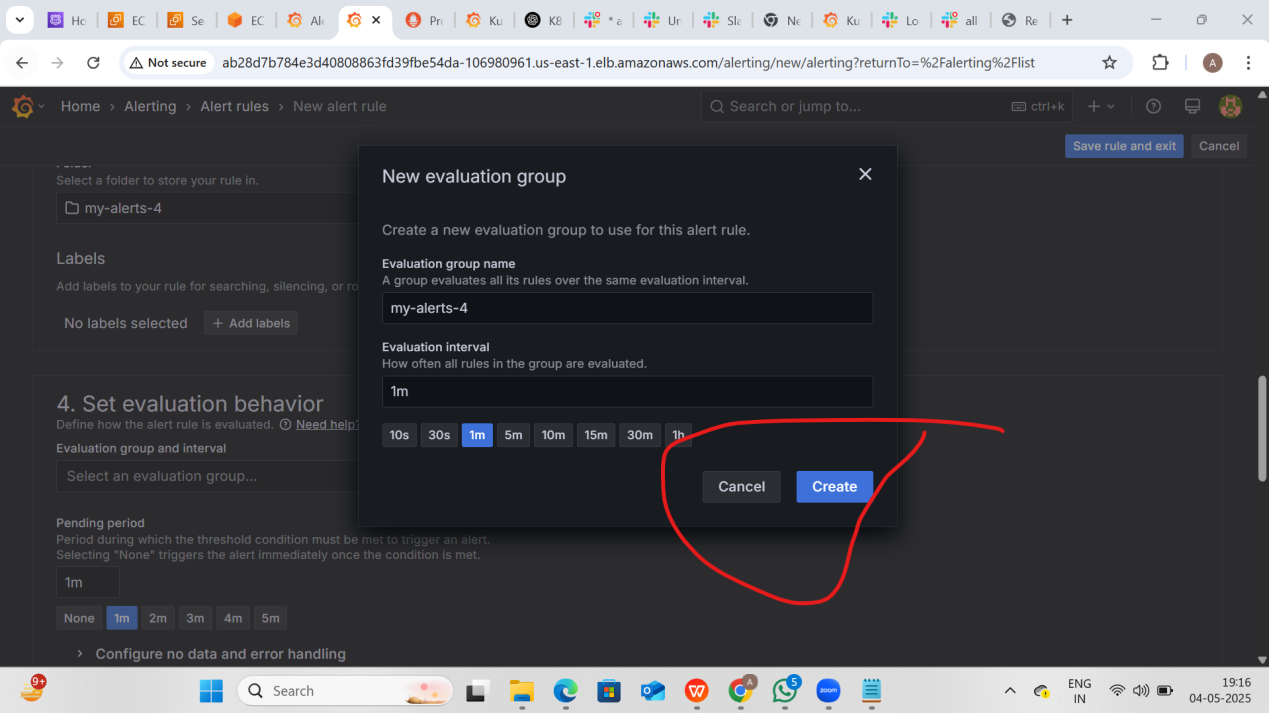


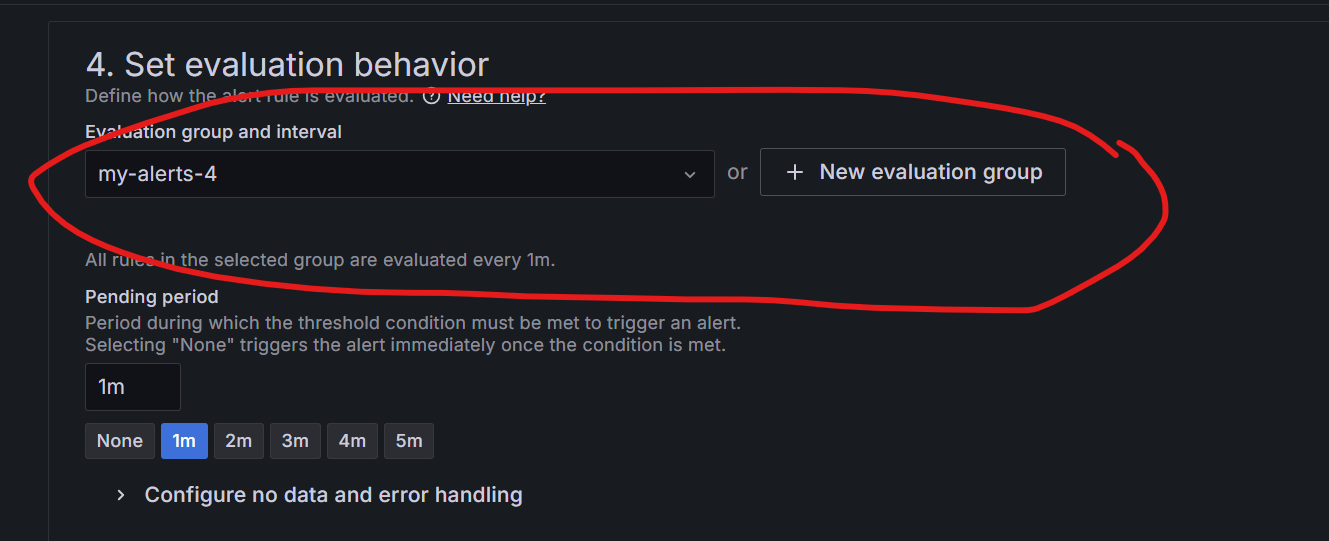
Click on **create**

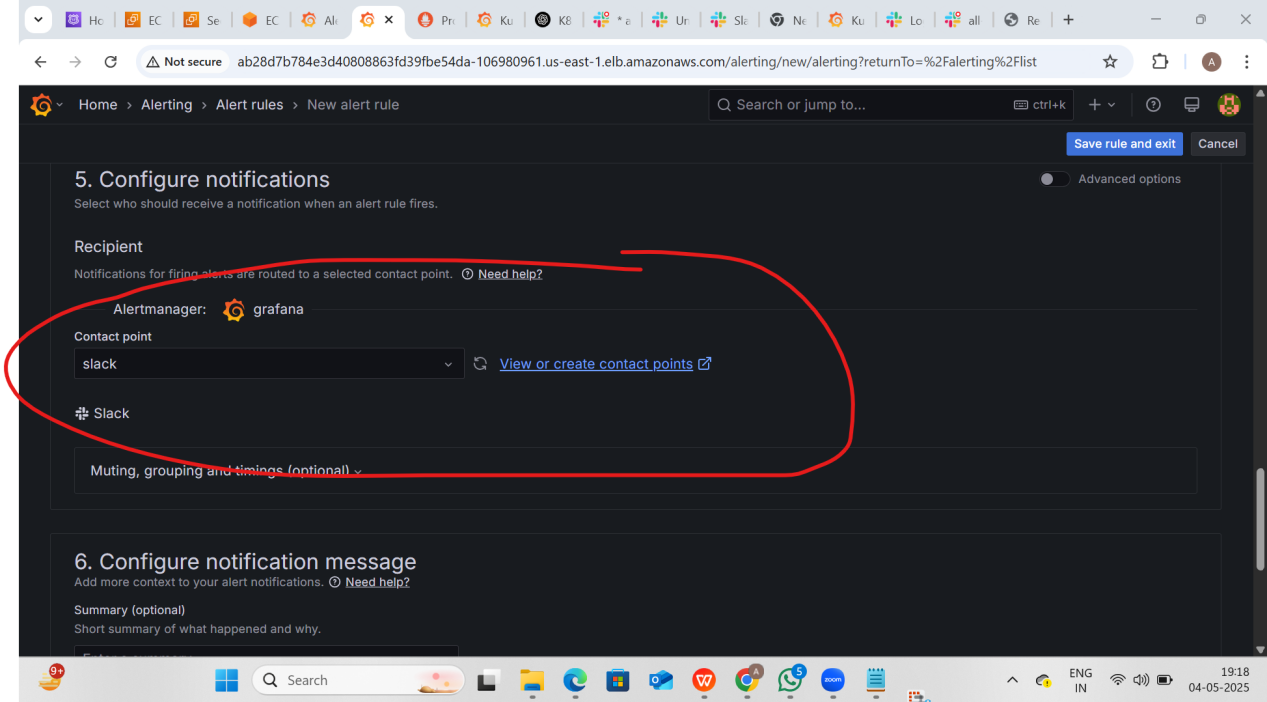




Click on **new evalution group**







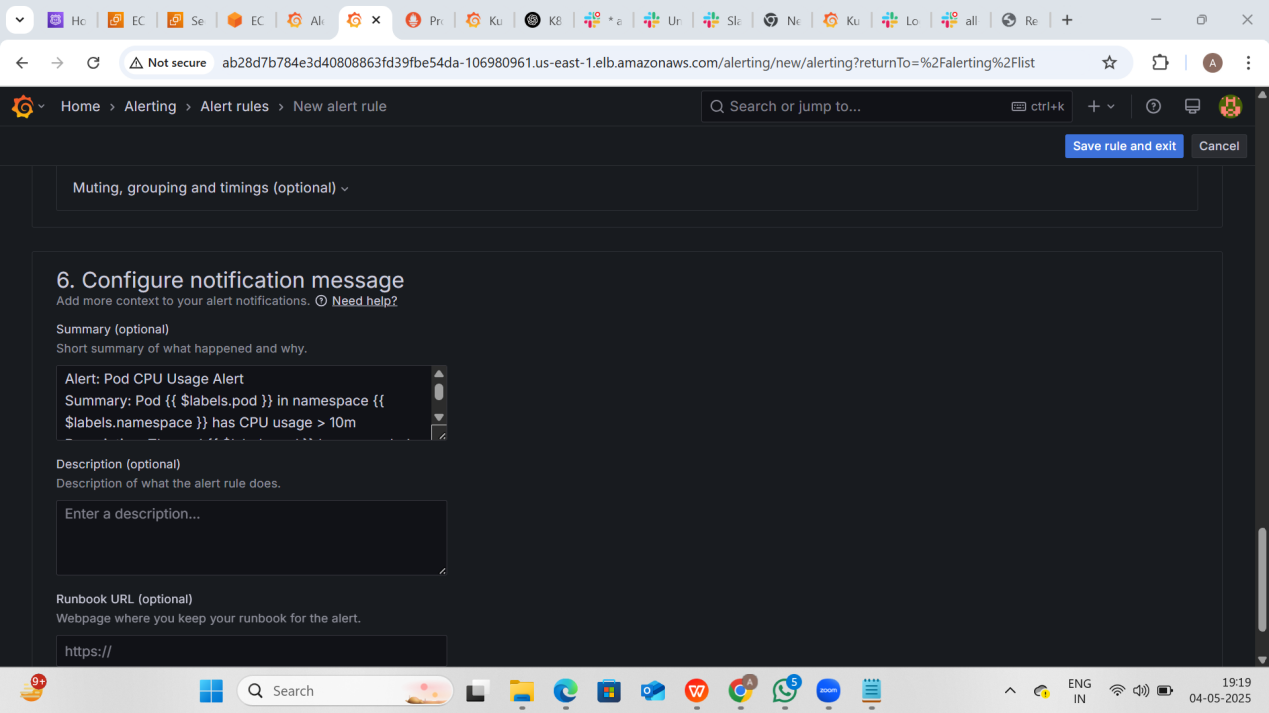
Select **slack**

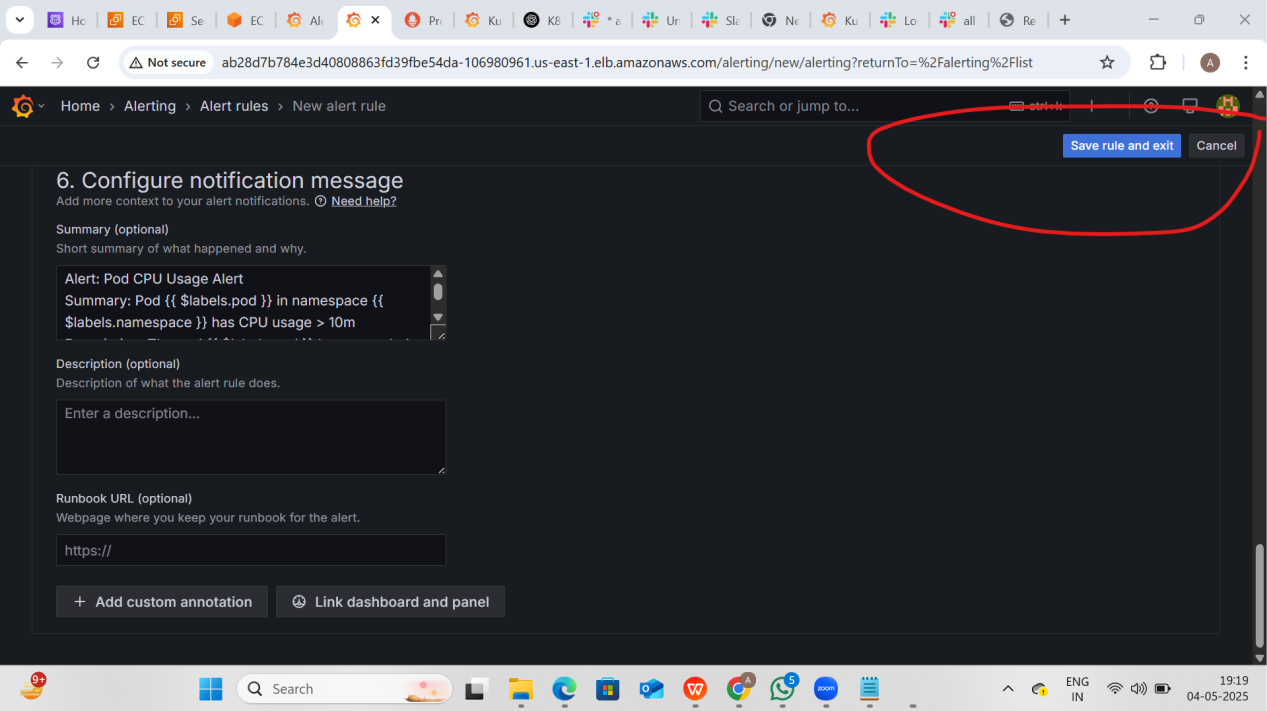
**Paste below summary**

**Alert: Pod CPU Usage Alert**

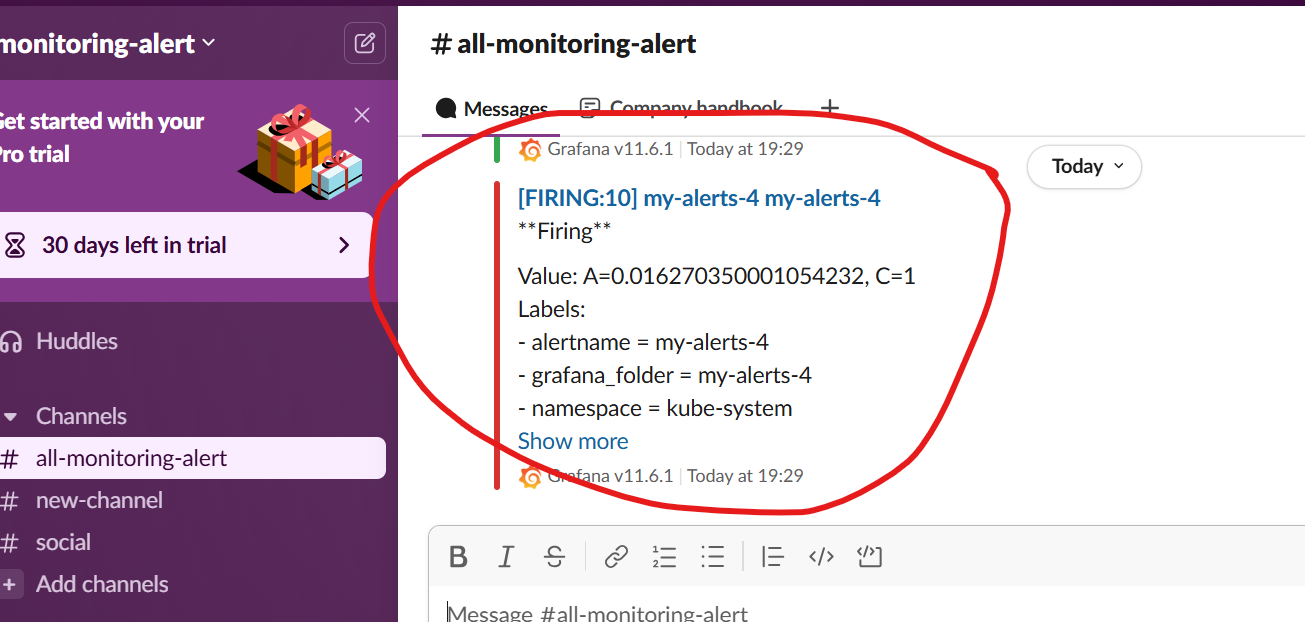
**Summary: Pod {{ $labels.pod }} in namespace {{ $labels.namespace }} has CPU usage > 10m**

**Description: The pod {{ $labels.pod }} has exceeded the CPU threshold.**

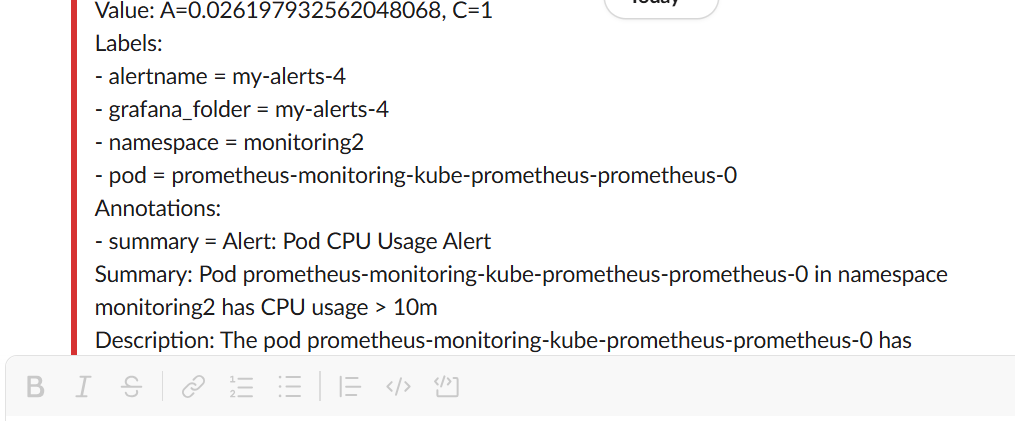




Save rule and exit



My-alert-4 is recevied



✅ **CPU Query (Exceeds 10m with Pod Name)**

sum by (pod) (rate(container\_cpu\_usage\_seconds\_total{container!="",pod!="",namespace!="", container!="POD"}[1m])) > 0.01

✅ **Memory Query (Exceeds 10Mi with Pod Name)**

**sum by (pod) (container\_memory\_usage\_bytes{container!="",pod!="",namespace!="", container!="POD"}) / 1024 / 1024 > 10**

**Use above PromQL Query and then u will receive slack notification automatically in your slack account for above conditions**

count(kube\_pod\_container\_status\_waiting\_reason{reason="ImagePullBackOff"})

kube\_pod\_container\_status\_waiting\_reason{reason="ImagePullBackOff"}

count(kube\_pod\_container\_status\_waiting\_reason{reason="ImagePullBackOff"})

kube\_pod\_container\_status\_waiting\_reason{reason="CrashLoopBackOff"} > 0