- 1, Creating EKS cluster with 2 worker nodes in your own VPC (see helm-project-instruction.pdf)
- 2, Depoloying ChatApp application using helm chart VPC (see helm-project-instruction.pdf)

3, Installing Kubernetes Metrics Server

Run command:

kubectl apply –f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml

```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W13\Day3\helm_project\charts> kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml
serviceaccount/metrics-server created
clusterrole.rbac.authorization.k8s.io/system:aggregated-metrics-reader created
clusterrole.rbac.authorization.k8s.io/system:metrics-server created
rolebinding.rbac.authorization.k8s.io/metrics-server-auth-reader created
clusterrolebinding.rbac.authorization.k8s.io/metrics-server:system:auth-delegator created
clusterrolebinding.rbac.authorization.k8s.io/system:metrics-server created
service/metrics-server created
deployment.apps/metrics-server created
apiservice.apiregistration.k8s.io/v1beta1.metrics.k8s.io created
```

```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W13\Day3\helm_project\charts> kubectl get deployment metrics-server -n kube-system
NAME             READY   UP-TO-DATE   AVAILABLE   AGE
metrics-server   1/1     1            1           100s
```

Check the resource utilization:

```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W13\Day3\helm_project\charts> kubectl top node
NAME                                           CPU(cores)   CPU%   MEMORY(bytes)   MEMORY%
ip-10-0-1-106.us-west-2.compute.internal   40m          4%     533Mi           35%
ip-10-0-2-182.us-west-2.compute.internal   53m          5%     1009Mi          67%
```

- 4, Deploying Prometheus on EKS Kubernetes Cluster

Step 1: Create Prometheus namespace:

Run command: kubectl create namespace Prometheus

```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W13\Day3\helm_project\charts> kubectl create namespace prometheus
namespace/prometheus created
```

Step 2: install

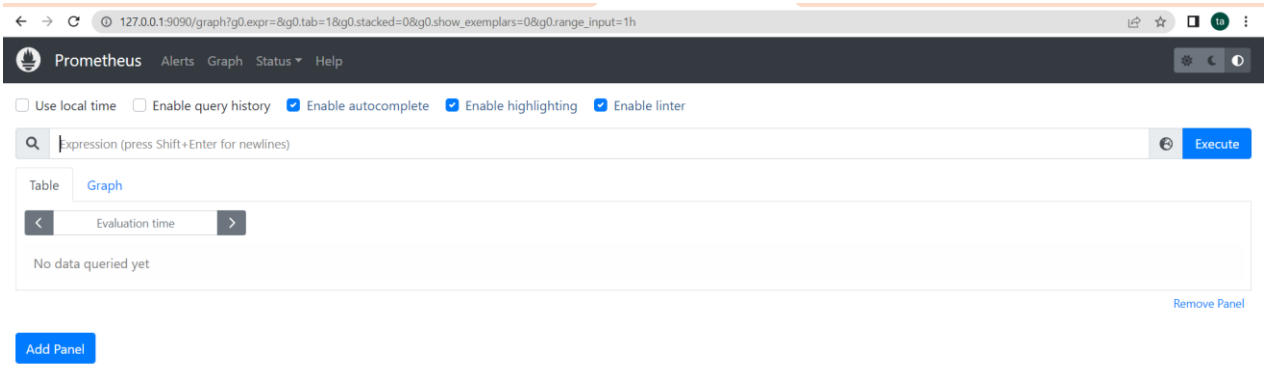
```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W13\Day3\helm_project\charts> helm install prometheus prometheus-community/prometheus --namespace prometheus --set alertmanager.persistentVolume.storageClass="gp2" --set server.persistentVolume.storageClass="gp2"
NAME: prometheus
LAST DEPLOYED: Sun Sep  4 15:51:39 2022
NAMESPACE: prometheus
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
The Prometheus server can be accessed via port 80 on the following DNS name from within your cluster:
prometheus-server.prometheus.svc.cluster.local


Get the Prometheus server URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace prometheus -l "app=prometheus,component=server" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace prometheus port-forward $POD_NAME 9090


The Prometheus alertmanager can be accessed via port 80 on the following DNS name from within your cluster:
prometheus-alertmanager.prometheus.svc.cluster.local


Get the Alertmanager URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace prometheus -l "app=prometheus,component=alertmanager" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace prometheus port-forward $POD_NAME 9093
#################################################################################
######   WARNING: Pod Security Policy has been moved to a global property.  #####
######            use .Values.podSecurityPolicy.enabled with pod-based      #####
######            annotations                                               #####
######            (e.g. .Values.nodeExporter.podSecurityPolicy.annotations) #####
#################################################################################


The Prometheus PushGateway can be accessed via port 9091 on the following DNS name from within your cluster:
prometheus-pushgateway.prometheus.svc.cluster.local


Get the PushGateway URL by running these commands in the same shell:
  export POD_NAME=$(kubectl get pods --namespace prometheus -l "app=prometheus,component=pushgateway" -o jsonpath="{.items[0].metadata.name}")
  kubectl --namespace prometheus port-forward $POD_NAME 9091
```

Check if Prometheus components deployed as expected

```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W13\Day3\helm_project\charts> kubectl get pod -n prometheus
NAME                                            READY   STATUS    RESTARTS   AGE
prometheus-alertmanager-669c87db47-jjsjm        2/2     Running   0          12m
prometheus-kube-state-metrics-77ddf69b4-lmw9l   1/1     Running   0          12m
prometheus-node-exporter-ljl2r                  1/1     Running   0          12m
prometheus-node-exporter-w764t                  1/1     Running   0          12m
prometheus-pushgateway-5c989b8998-tmjf7         1/1     Running   0          12m
prometheus-server-57f4946cdb-hpnjn              2/2     Running   0          12m
```

Get pod forward:

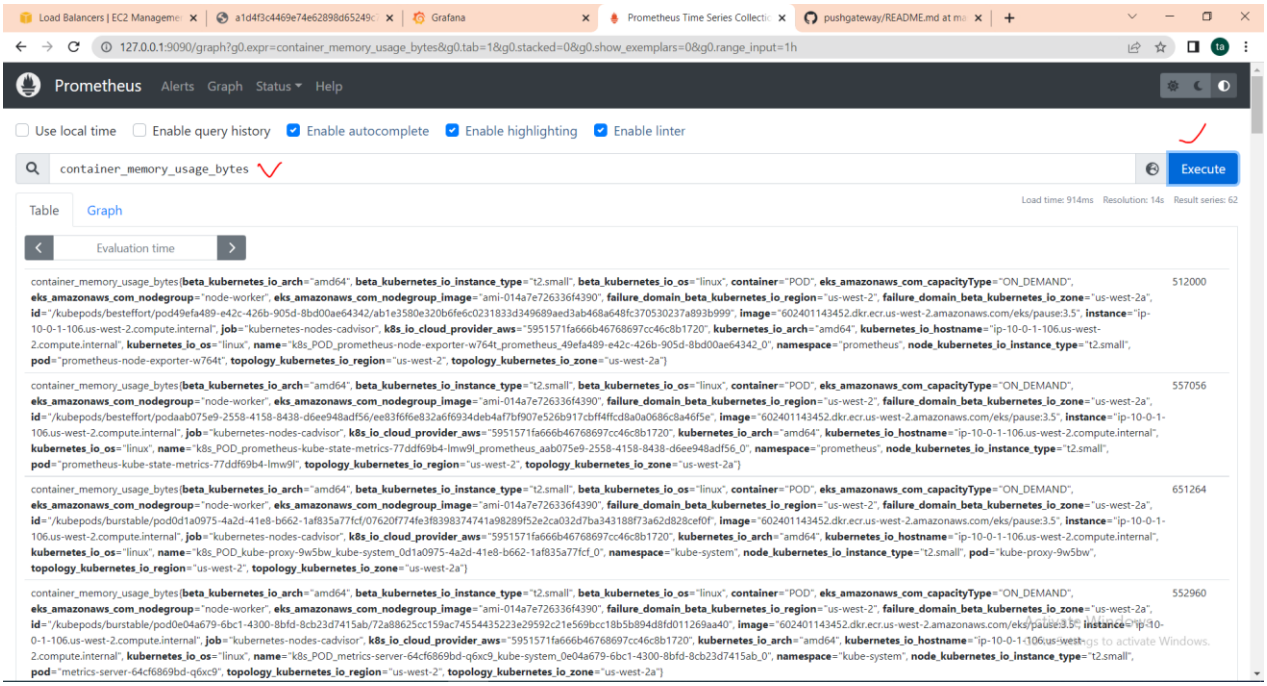
```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W13\Day3\helm_project\charts> kubectl --namespace prometheus port-forward prometheus-server-57f4946cdb-hpnj
n 9090
Forwarding from 127.0.0.1:9090 -> 9090
Forwarding from [::1]:9090 -> 9090
Handling connection for 9090
Handling connection for 9090
Handling connection for 9090
```

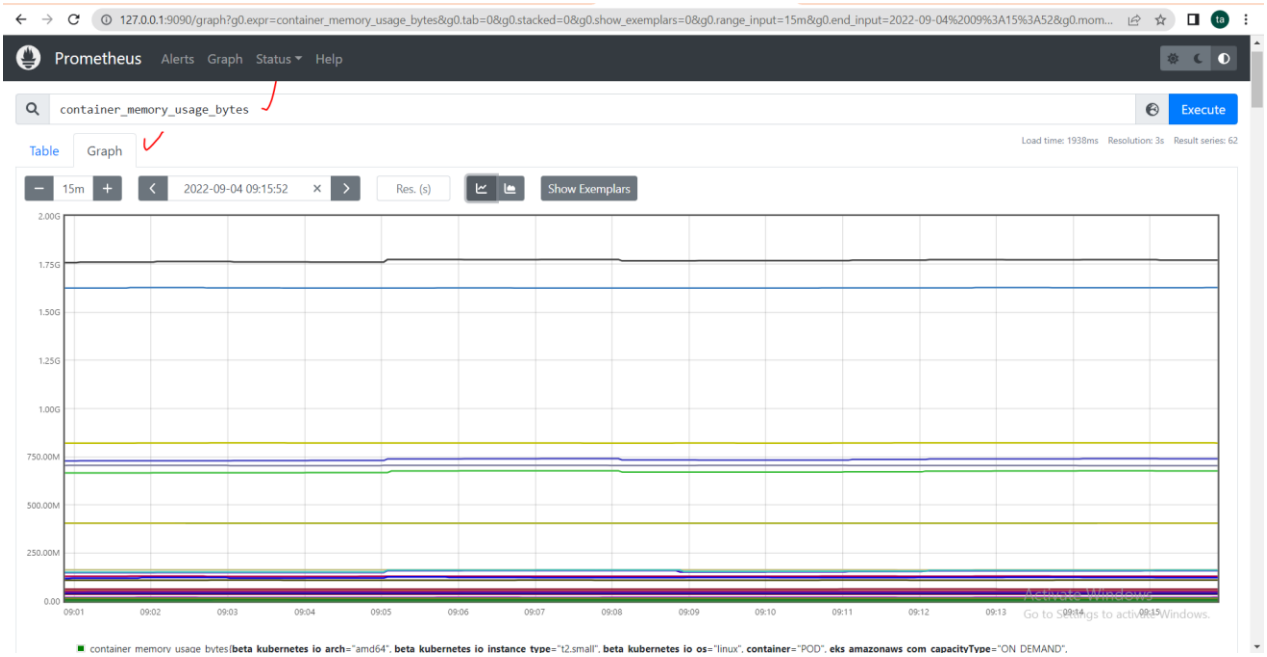



- 5, Monitoring EKS cluster using Prometheus

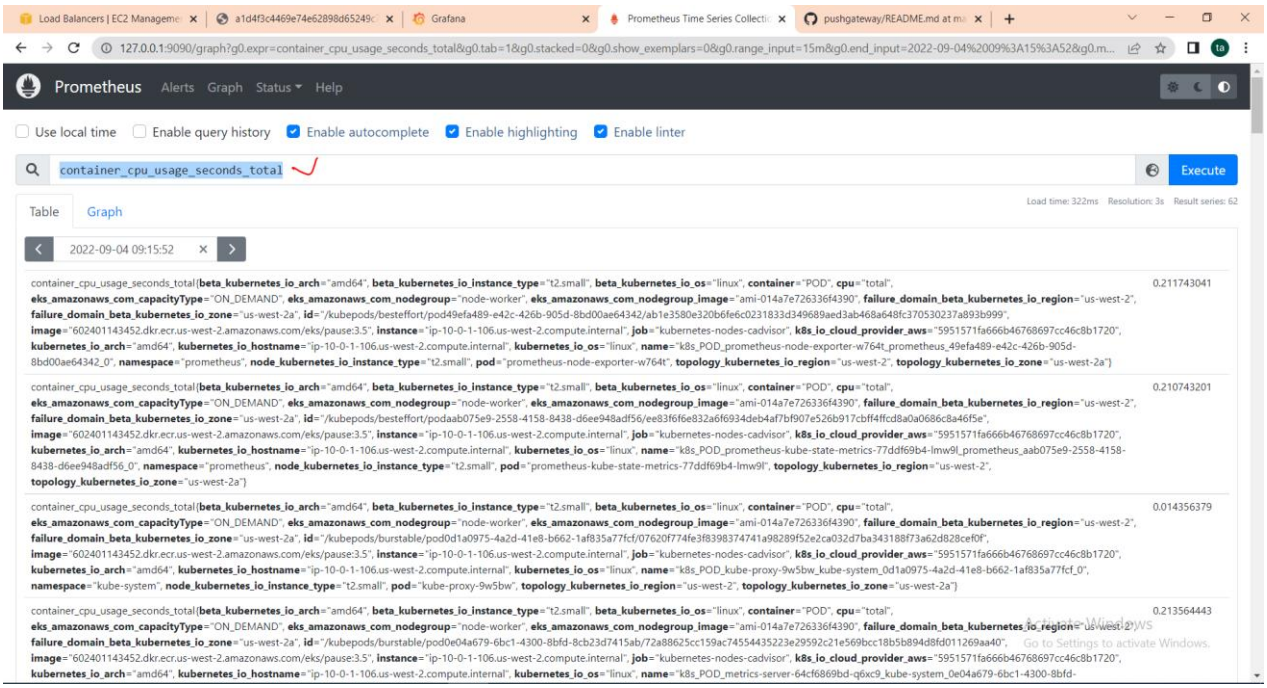
  6, Querying some metrics of EKS cluster, such as: CPU, memory, network latency, disk utilization

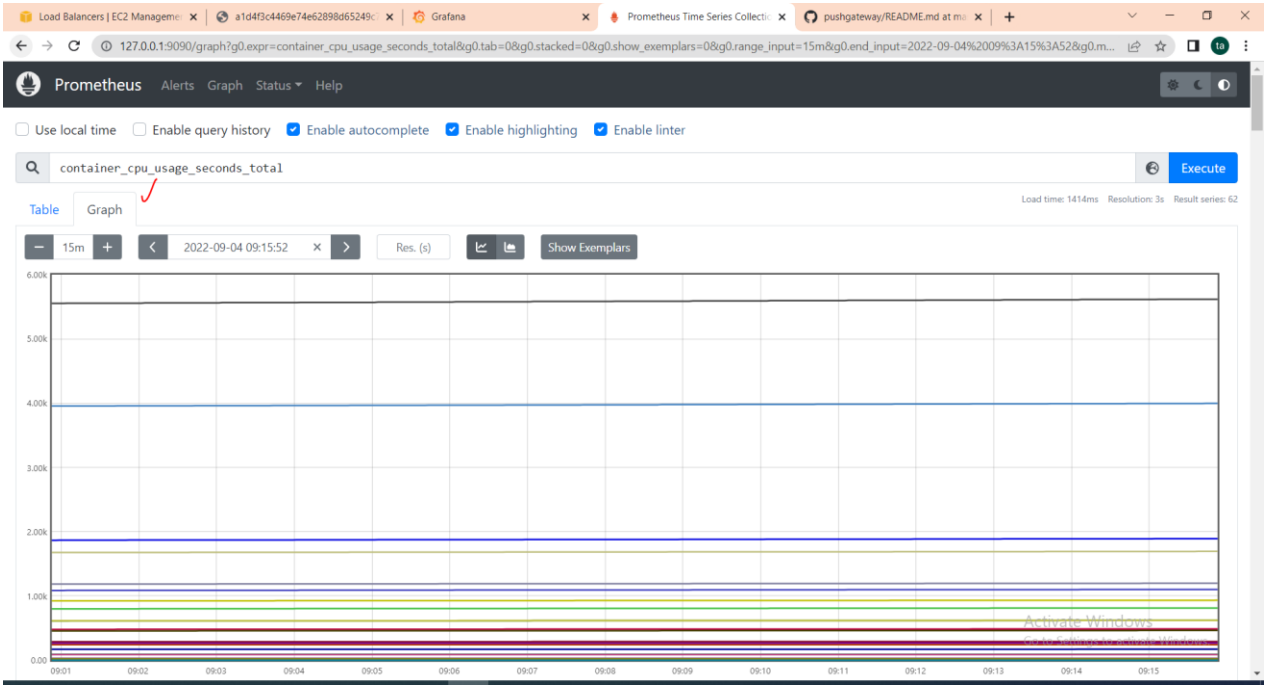
  - query container_memory_usage_bytes:



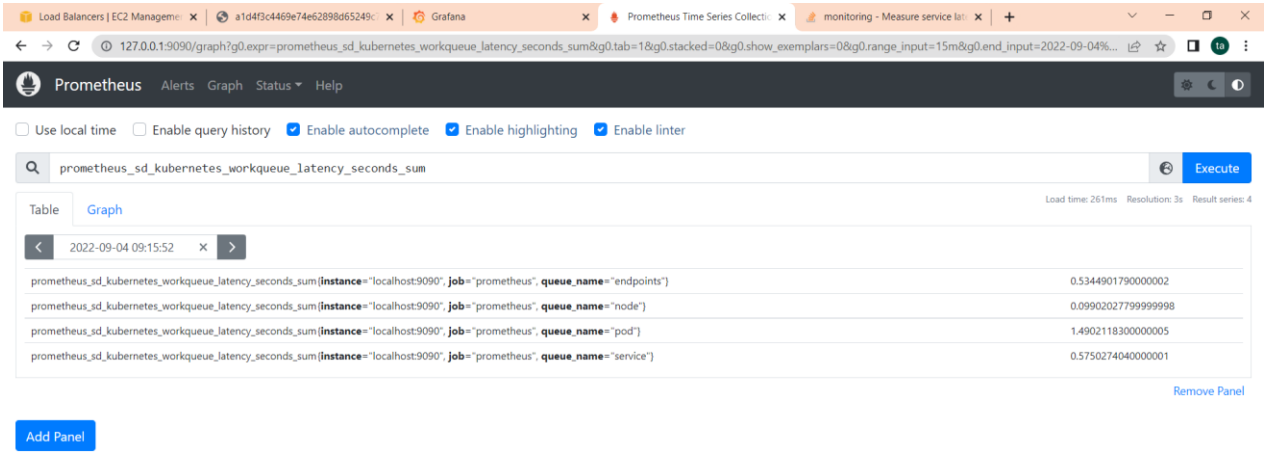


- query : “container_cpu_usage_seconds_total”

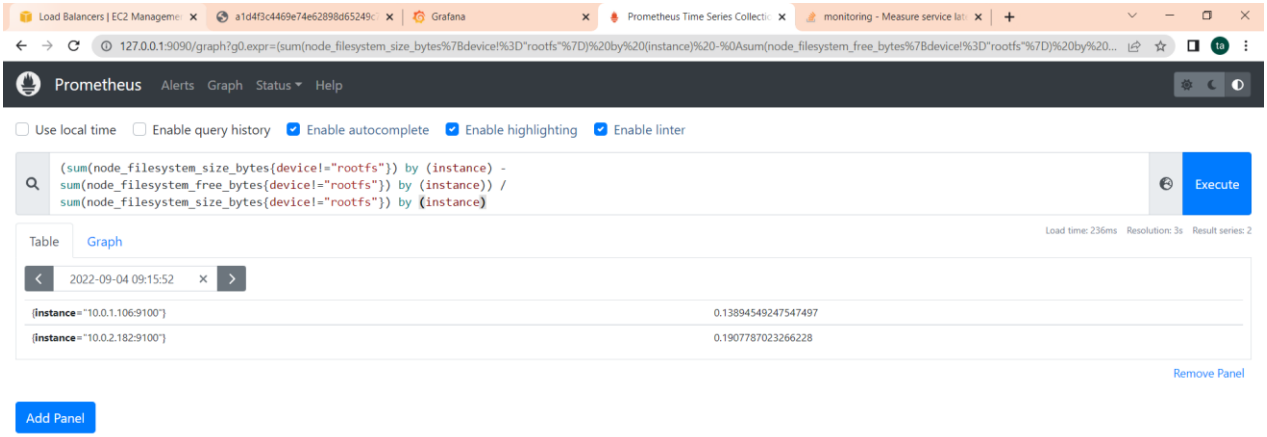




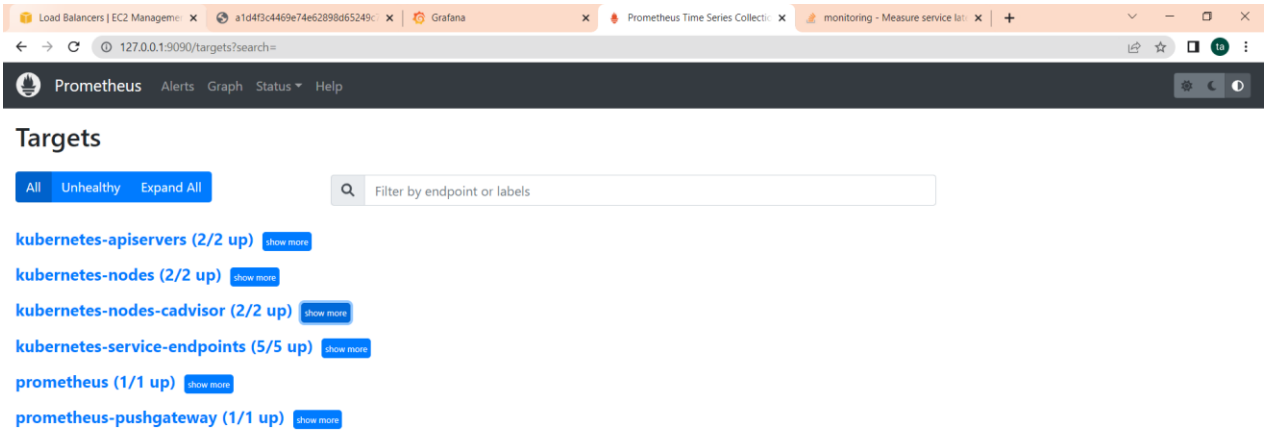
- query: “network latency”



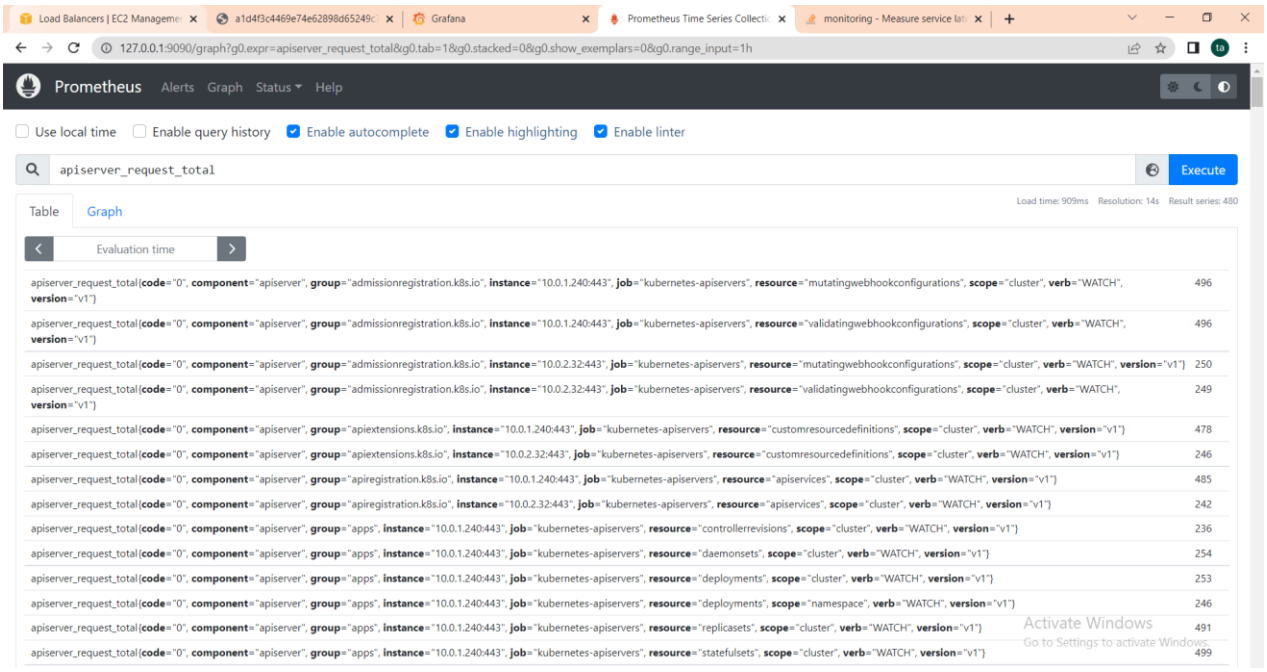
- query: “disk utilization”



- 7, Showing result ChatApp working and log from Prometheus



Check: “apiserver_request_total”



Monitoring with grafana:

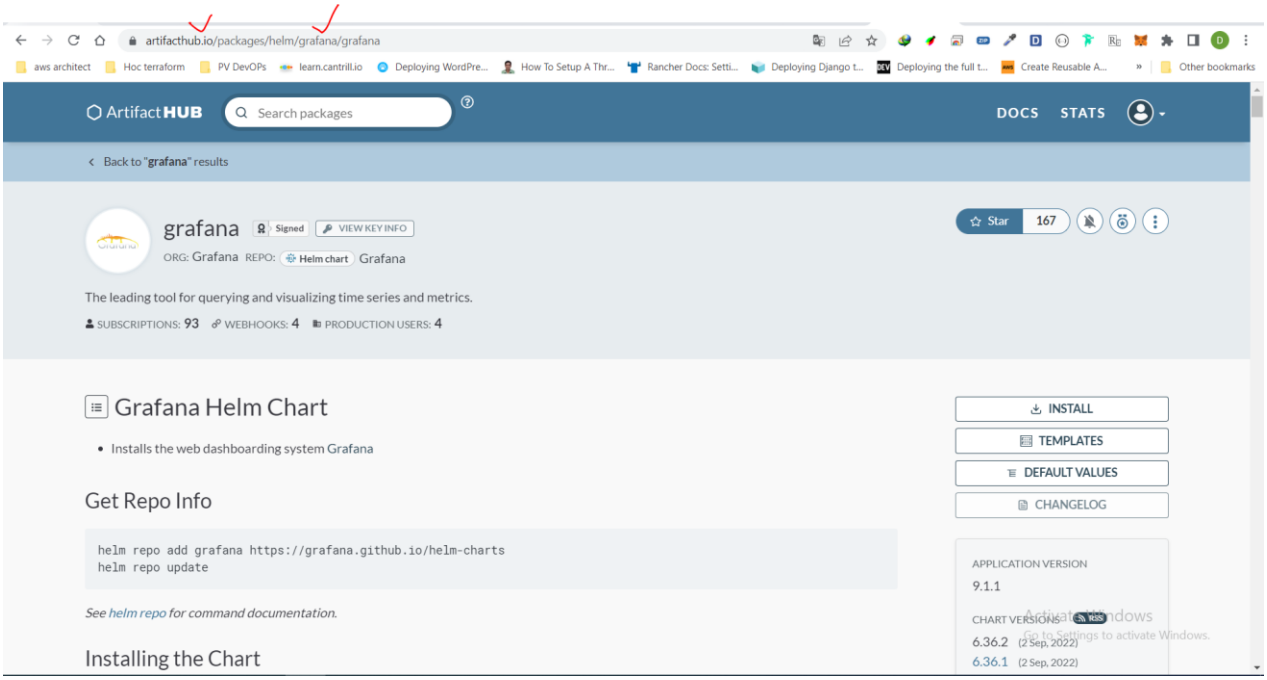
Step 1:

Create namespace Grafana:

```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W13\Day3\helm_project\charts> kubectl create namespace grafana
```

Step 2:

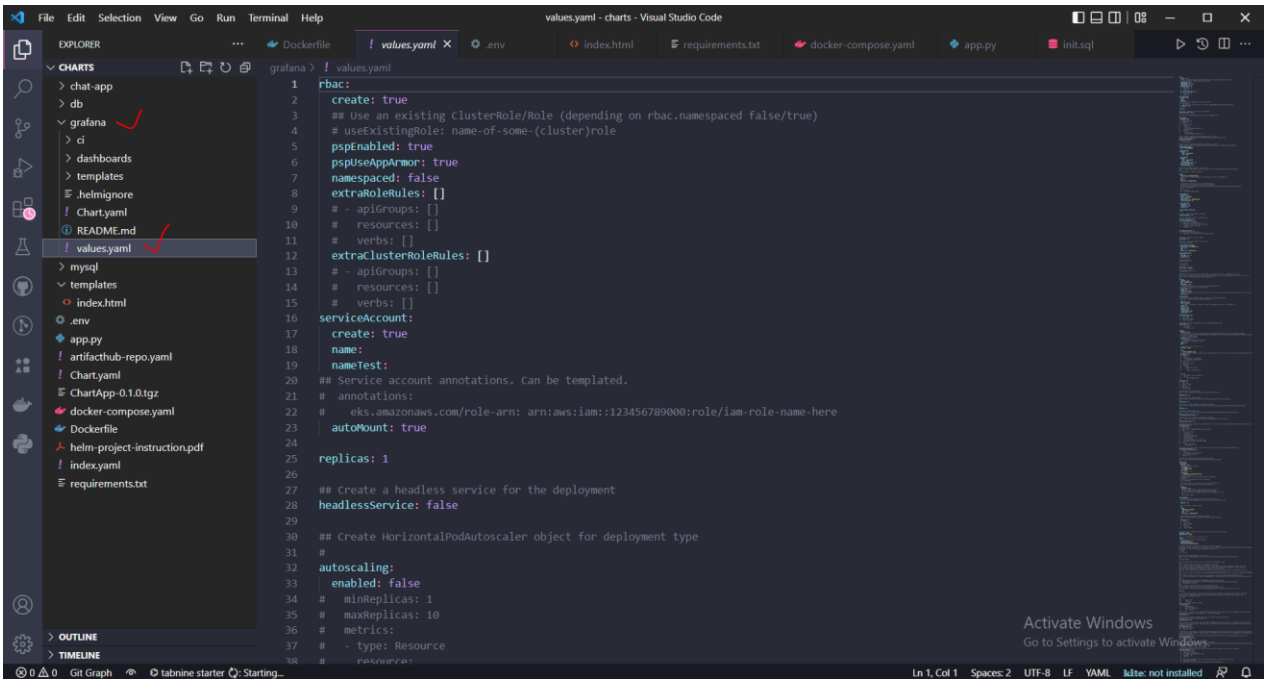
Add repo grafana:



Step 3:Customize file values.yaml in folder grafana repo:

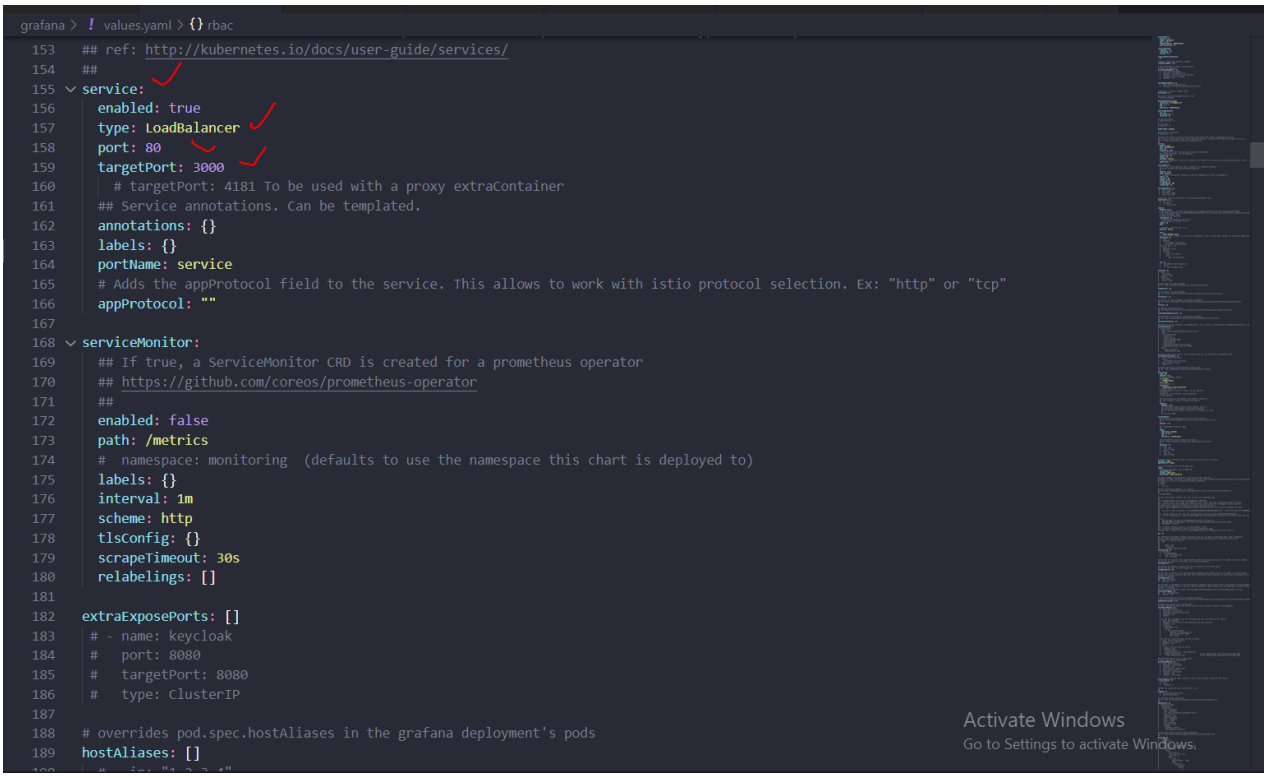
Step 1: open file values.yaml

```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W13\Day3\helm_project\charts> helm repo add grafana https://grafana.github.io/helm-charts
"grafana" has been added to your repositories
```



Step 2:

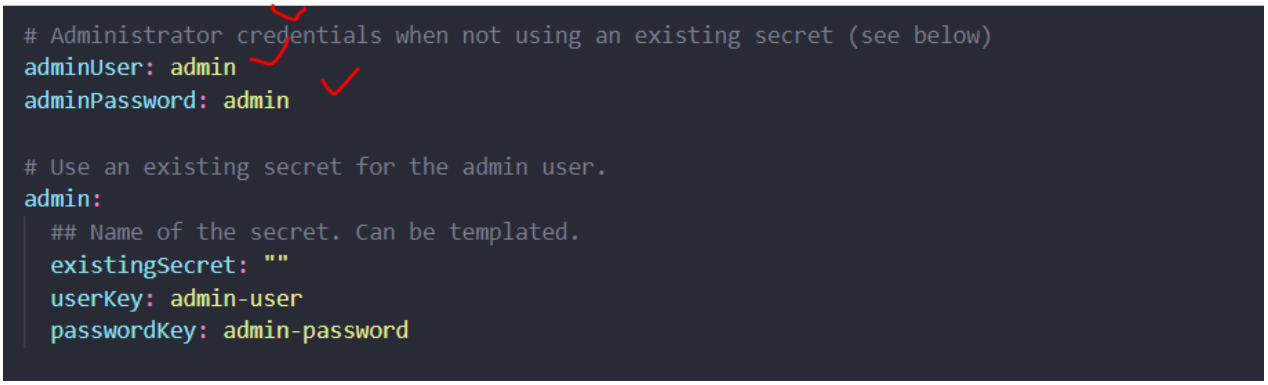
Configure service type: LoadBalancer



```
grafana > ! values.yaml > {} rbac
153   ## ref: http://kubernetes.io/docs/user-guide/services/
154   ##
155   service:
156     enabled: true
157     type: LoadBalancer
158     port: 80
159     targetPort: 3000
160       # targetPort: 4181 To be used with a proxy extraContainer
161     ## Service annotations. Can be templated.
162     annotations: {}
163     labels: {}
164     portName: service
165     # Adds the appProtocol field to the service. This allows to work with istio protocol selection. Ex: "http" or "tcp"
166     appProtocol: ""
167
168   serviceMonitor:
169     ## If true, a ServiceMonitor CRD is created for a prometheus operator
170     ## https://github.com/coreos/prometheus-operator
171     ##
172     enabled: false
173     path: /metrics
174     #  namespace: monitoring  (defaults to use the namespace this chart is deployed to)
175     labels: {}
176     interval: 1m
177     scheme: http
178     tlsConfig: {}
179     scrapeTimeout: 30s
180     relabelings: []
181
182   extraExposePorts: []
183     # - name: keycloak
184     #   port: 8080
185     #   targetPort: 8080
186     #   type: ClusterIP
187
188   # overrides pod.spec.hostAliases in the grafana deployment's pods
189   hostAliases: []
```


Step 3:

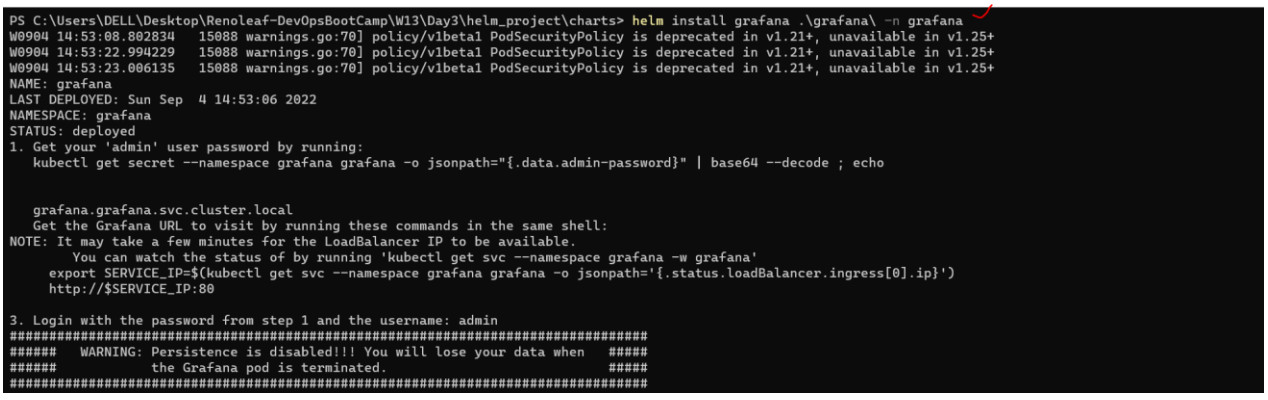
Configure user, password for admin



```
# Administrator credentials when not using an existing secret (see below)
adminUser: admin
adminPassword: admin

# Use an existing secret for the admin user.
admin:
  ## Name of the secret. Can be templated.
  existingSecret: ""
  userKey: admin-user
  passwordKey: admin-password
```

Step 4: Install grafana:

Run command:

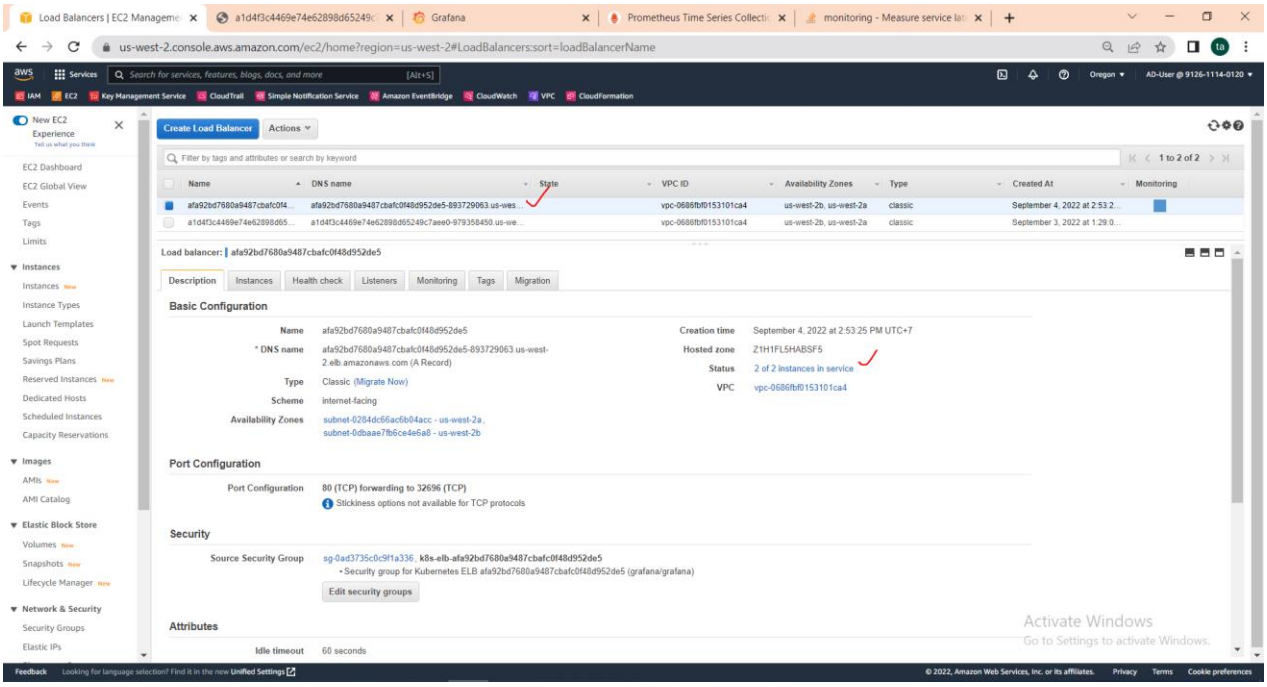


```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W13\Day3\helm_project\charts> helm install grafana .\grafana\ -n grafana
W0904 14:53:08.802834   15088 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W0904 14:53:22.994229   15088 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
W0904 14:53:23.006135   15088 warnings.go:70] policy/v1beta1 PodSecurityPolicy is deprecated in v1.21+, unavailable in v1.25+
NAME: grafana
LAST DEPLOYED: Sun Sep  4 14:53:06 2022
NAMESPACE: grafana
STATUS: deployed
1. Get your 'admin' user password by running:
   kubectl get secret --namespace grafana grafana -o jsonpath="{.data.admin-password}" | base64 --decode ; echo


   grafana.grafana.svc.cluster.local
   Get the Grafana URL to visit by running these commands in the same shell:
NOTE: It may take a few minutes for the LoadBalancer IP to be available.
      You can watch the status of by running 'kubectl get svc --namespace grafana -w grafana'
    export SERVICE_IP=$(kubectl get svc --namespace grafana grafana -o jsonpath='{.status.loadBalancer.ingress[0].ip}')
    http://$SERVICE_IP:80

3. Login with the password from step 1 and the username: admin
#################################################################################
######   WARNING: Persistence is disabled!!! You will lose your data when   #####
######            the Grafana pod is terminated.                            #####
#################################################################################
```

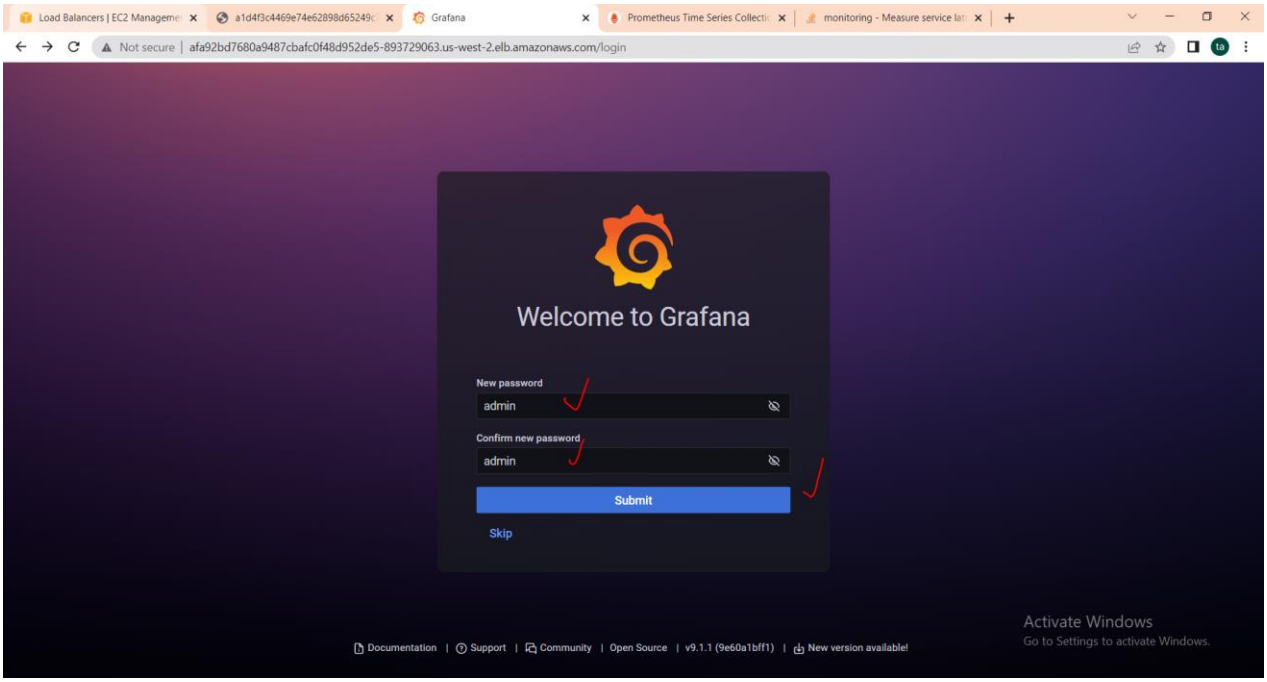
Check service grafana:

```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W13\Day3\helm_project\charts> kubectl get svc --namespace grafana -w grafana
NAME      TYPE           CLUSTER-IP     EXTERNAL-IP                                                              PORT(S)        AGE
grafana   LoadBalancer   172.20.37.27   afa92bd7680a9487cbafc0f48d952de5-893729063.us-west-2.elb.amazonaws.com   80:32696/TCP   40s
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W13\Day3\helm_project\charts> kubectl get pod,svc -n grafana
NAME                           READY   STATUS    RESTARTS   AGE
pod/grafana-fdd75664c-6vx28    1/1     Running   0          115s

NAME              TYPE           CLUSTER-IP     EXTERNAL-IP                                                              PORT(S)        AGE
service/grafana   LoadBalancer   172.20.37.27   afa92bd7680a9487cbafc0f48d952de5-893729063.us-west-2.elb.amazonaws.com   80:32696/TCP   116s
```

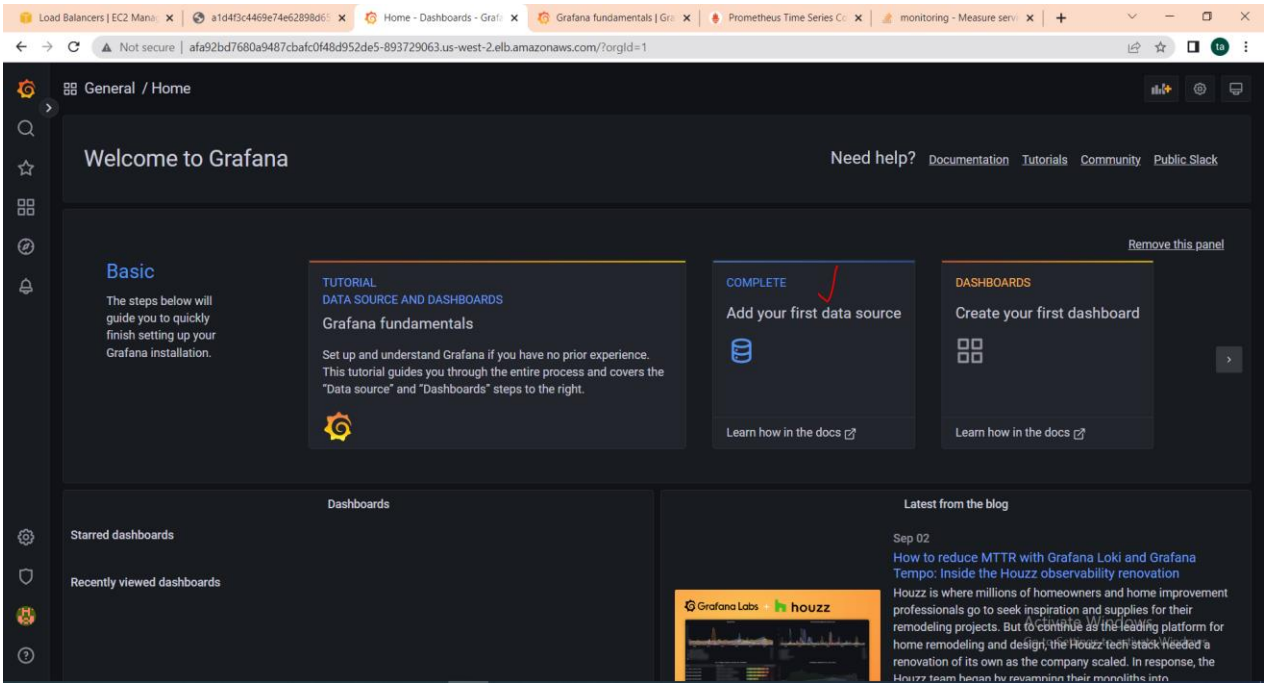
Check load balancer:

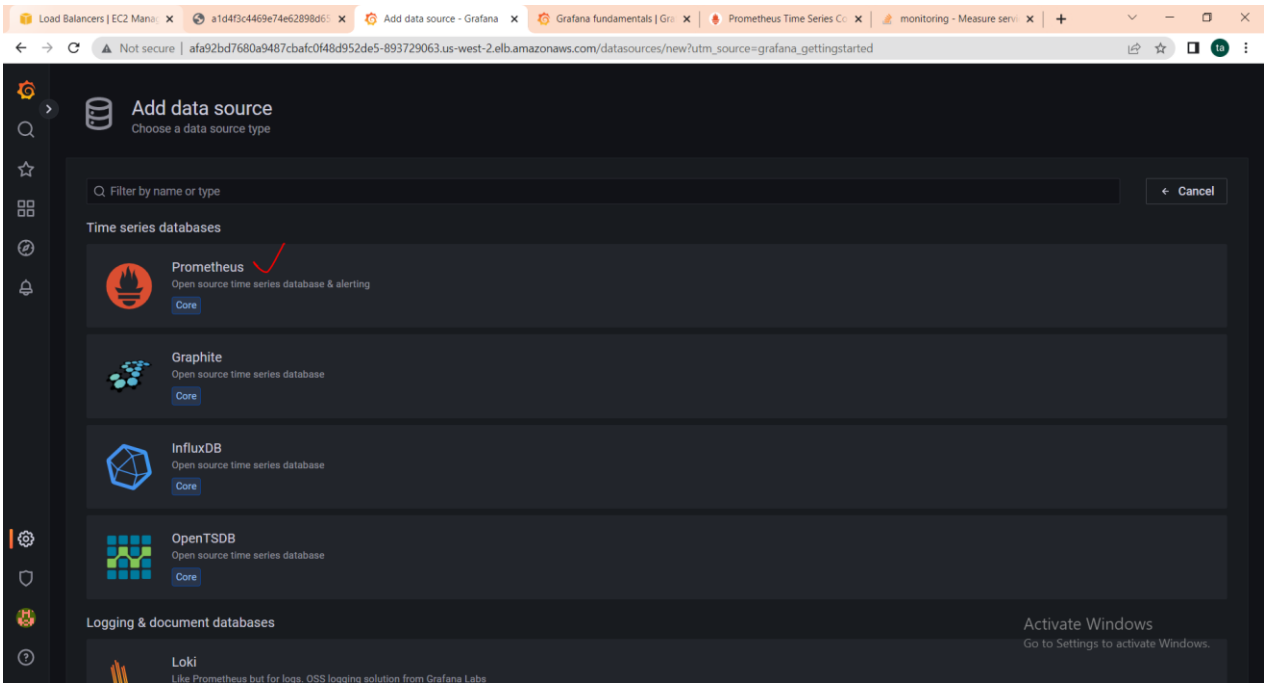


Step 4:



Step 5: Add data source:

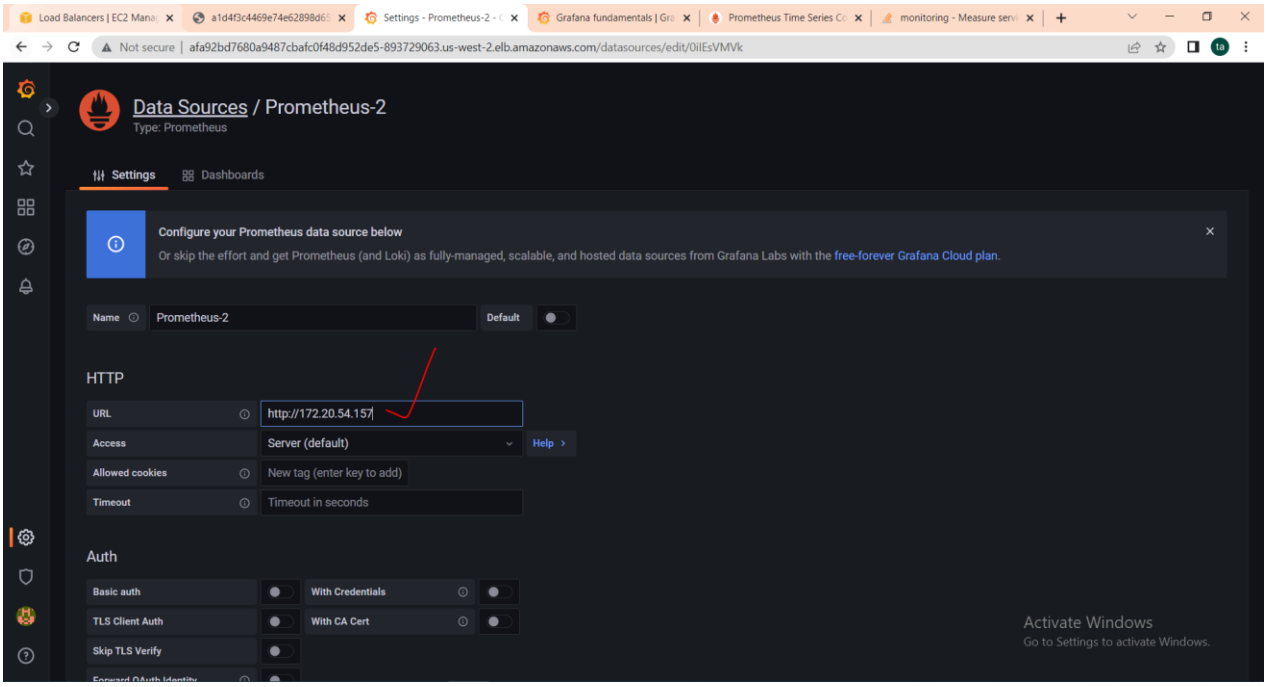


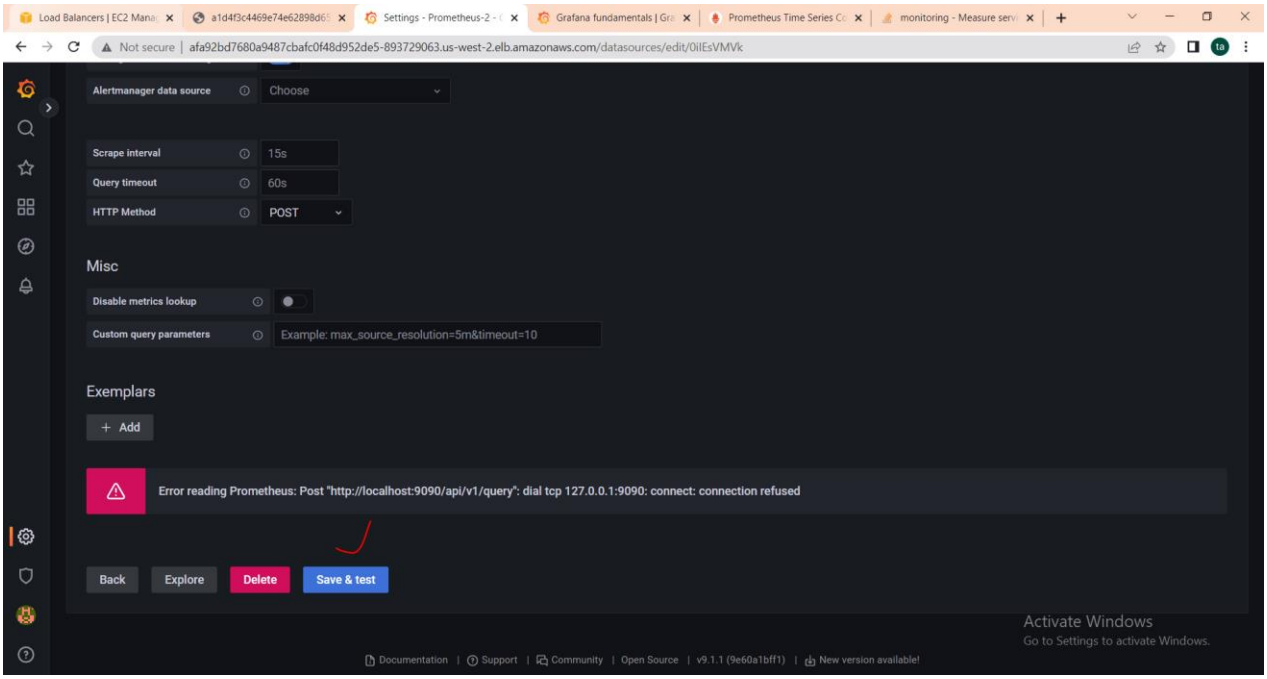


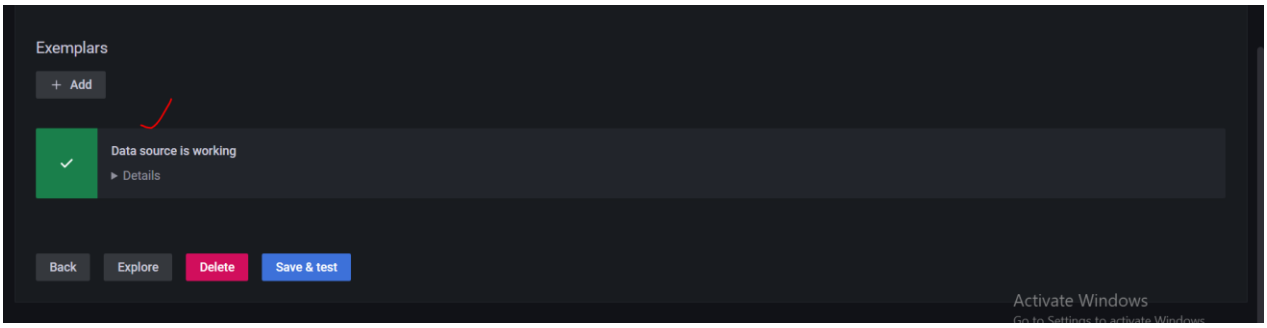
Check Prometheus service:

```
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W13\Day3\helm_project\charts> kubectl get svc -n prometheus
NAME                            TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)    AGE
prometheus-alertmanager         ClusterIP   172.20.248.223   <none>        80/TCP     76m
prometheus-kube-state-metrics   ClusterIP   172.20.252.181   <none>        8080/TCP   76m
prometheus-node-exporter        ClusterIP   172.20.197.90    <none>        9100/TCP   76m
prometheus-pushgateway          ClusterIP   172.20.147.7     <none>        9091/TCP   76m
prometheus-server               ClusterIP   172.20.54.157    <none>        80/TCP     76m
PS C:\Users\DELL\Desktop\Renoleaf-DevOpsBootCamp\W13\Day3\helm_project\charts>
```

Enter URL

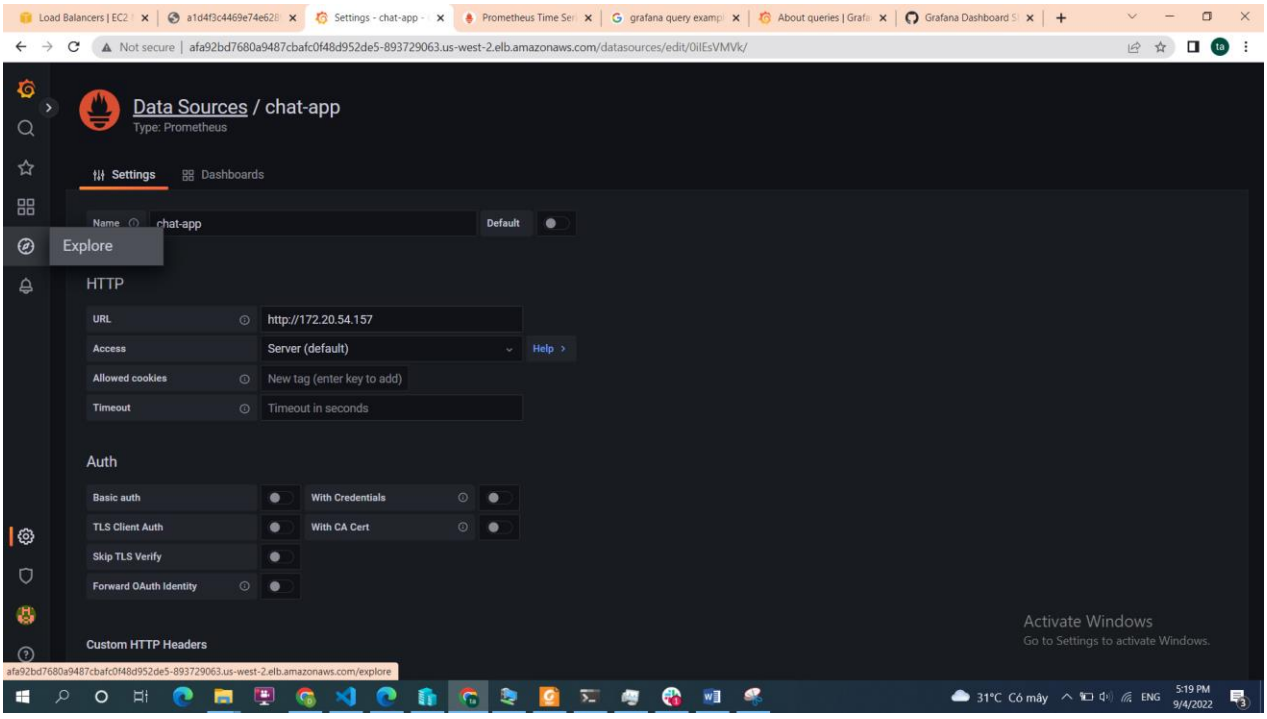


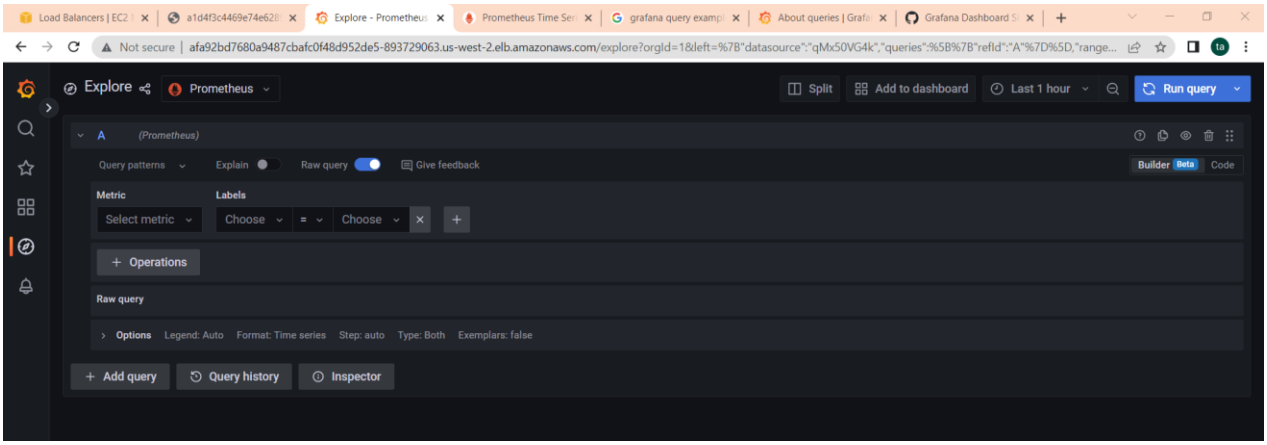




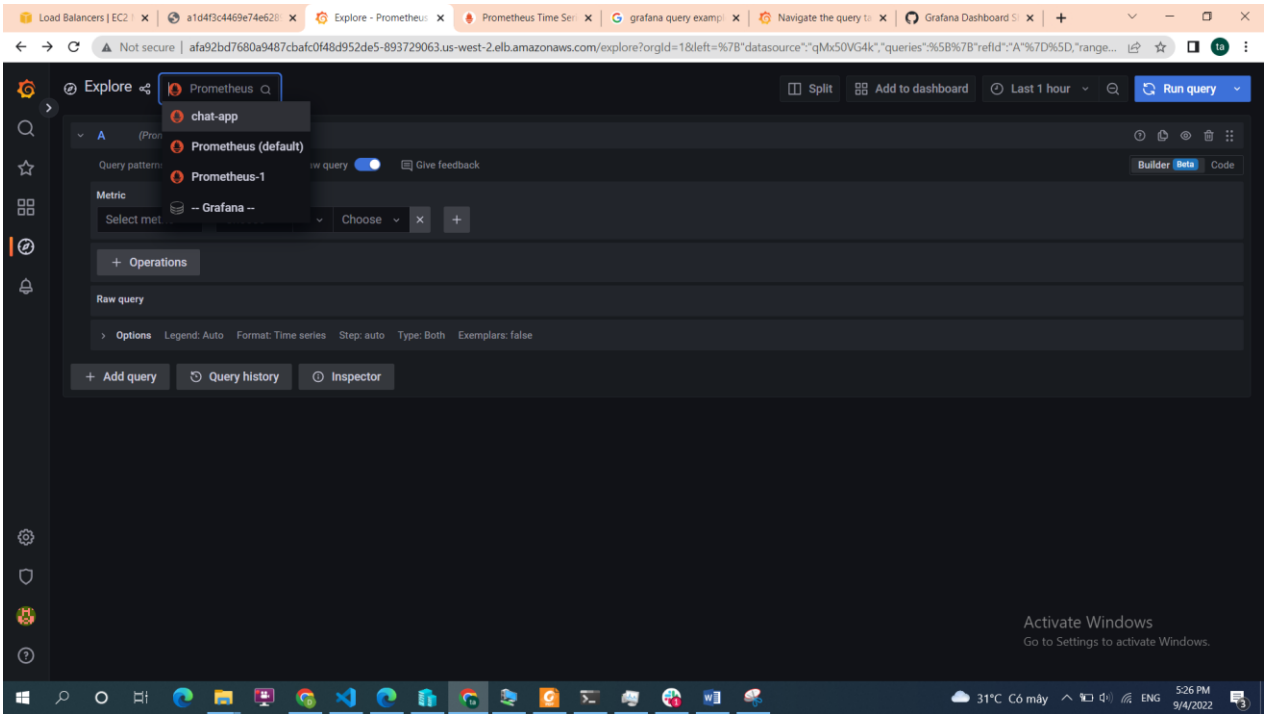
Query:

Go to tab explore:

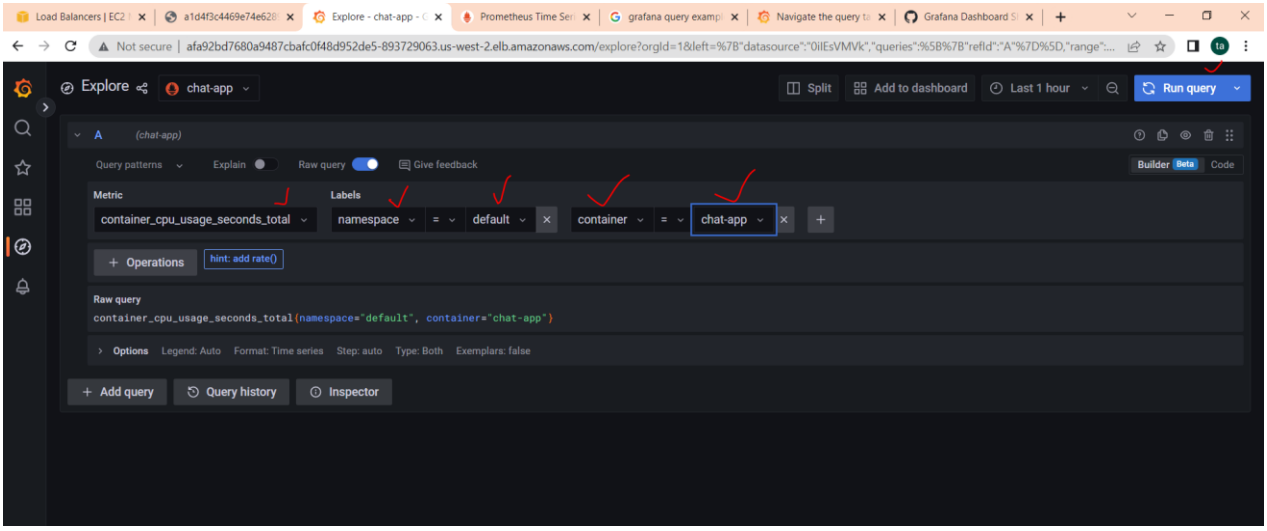




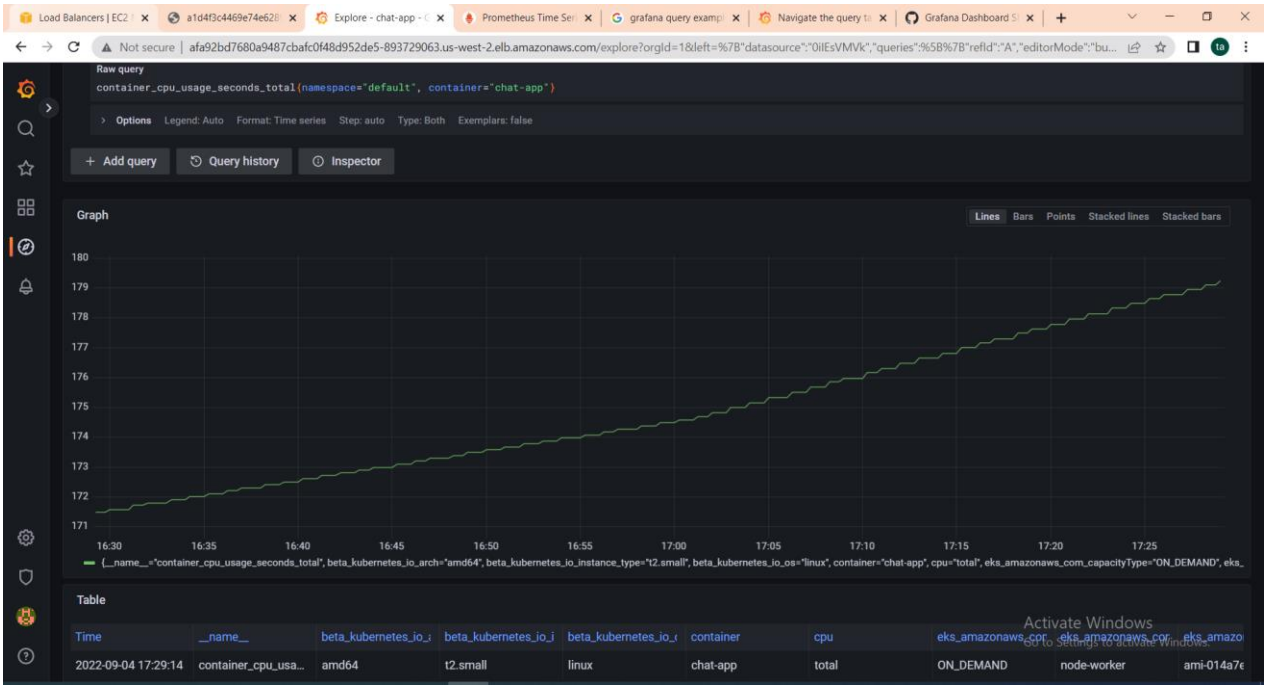
Select chat-app project to query:



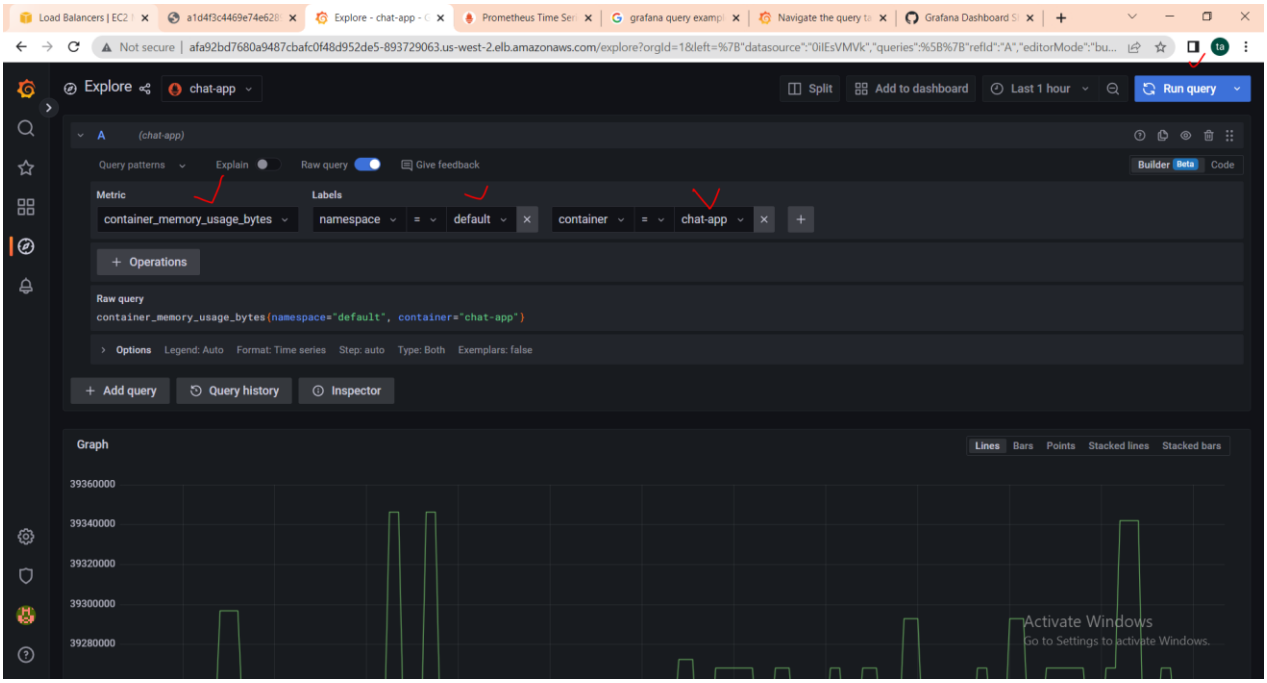
Query: CPU



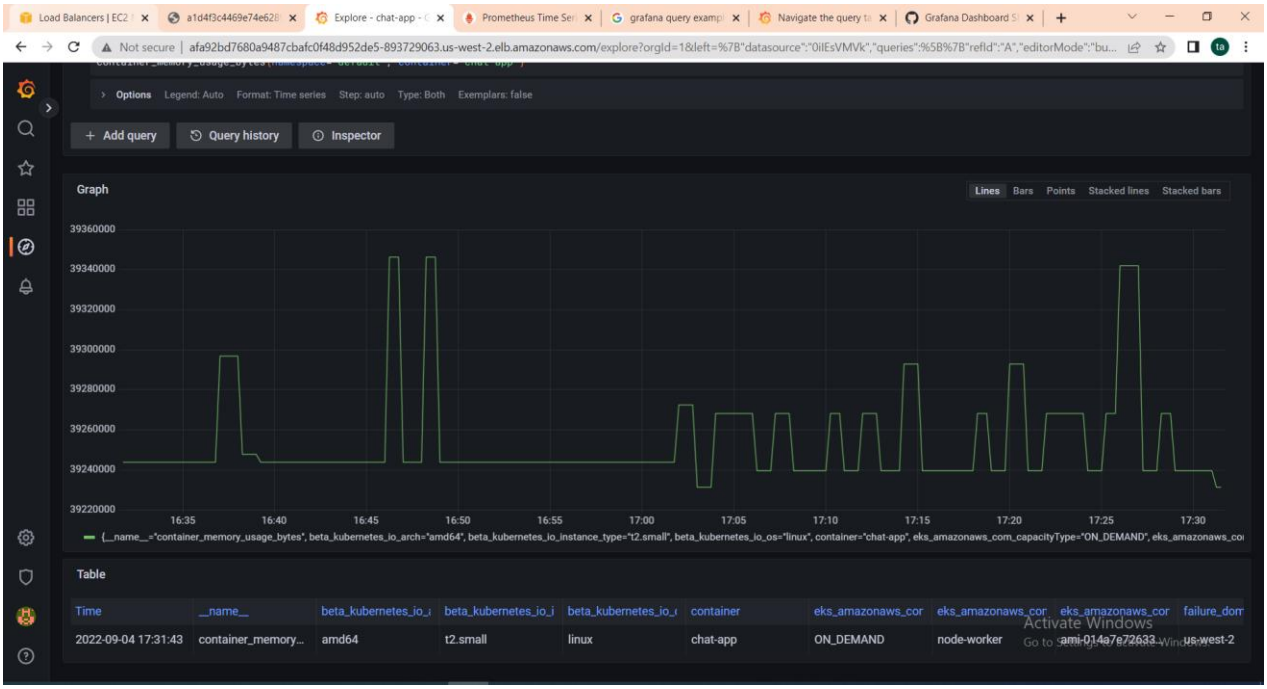
Result:



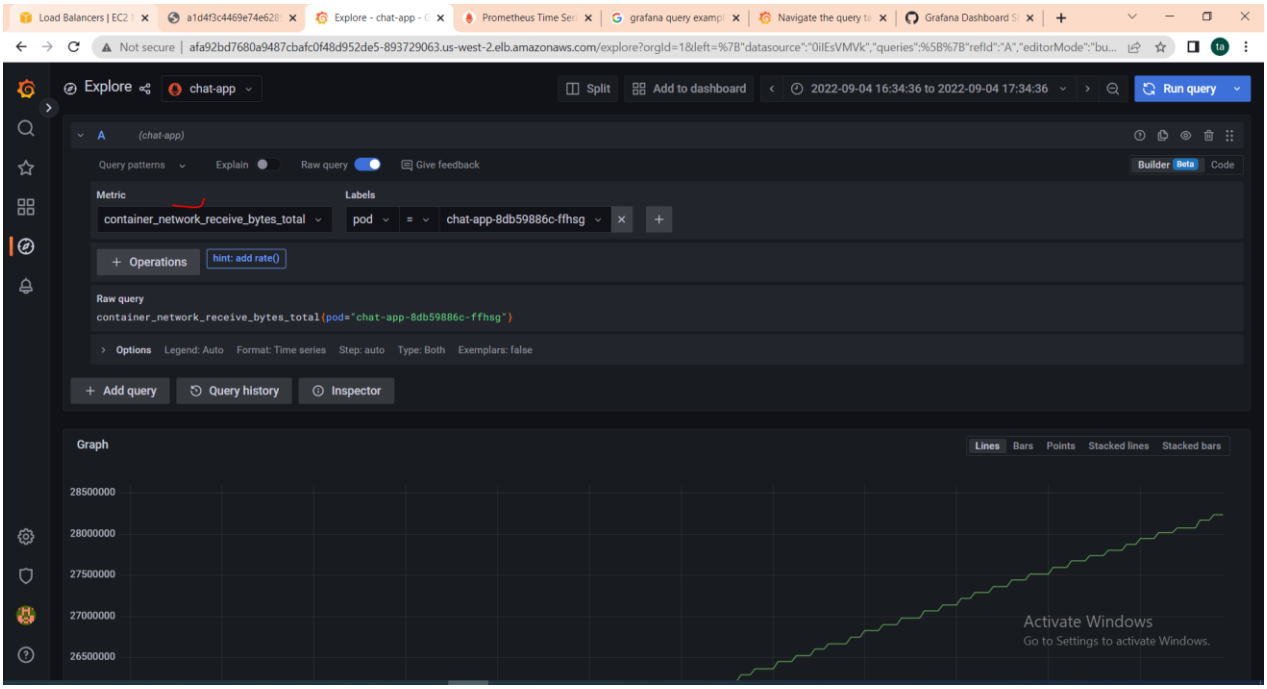
-Memory:



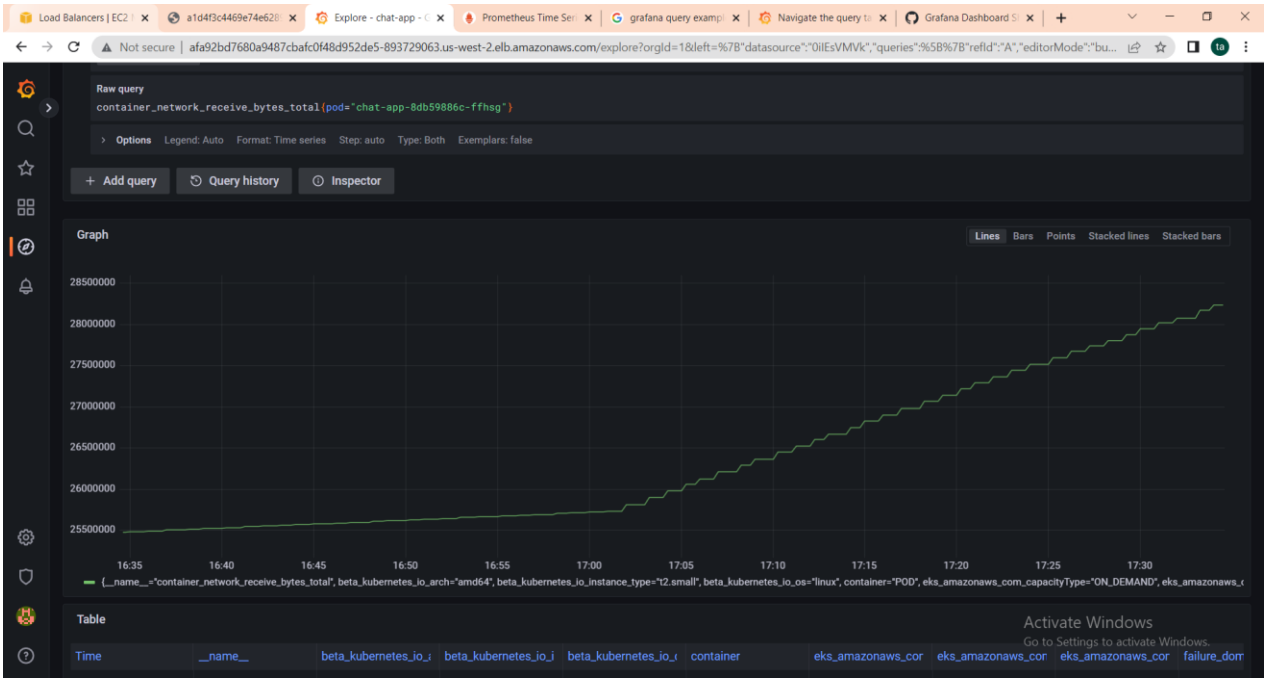
Result:



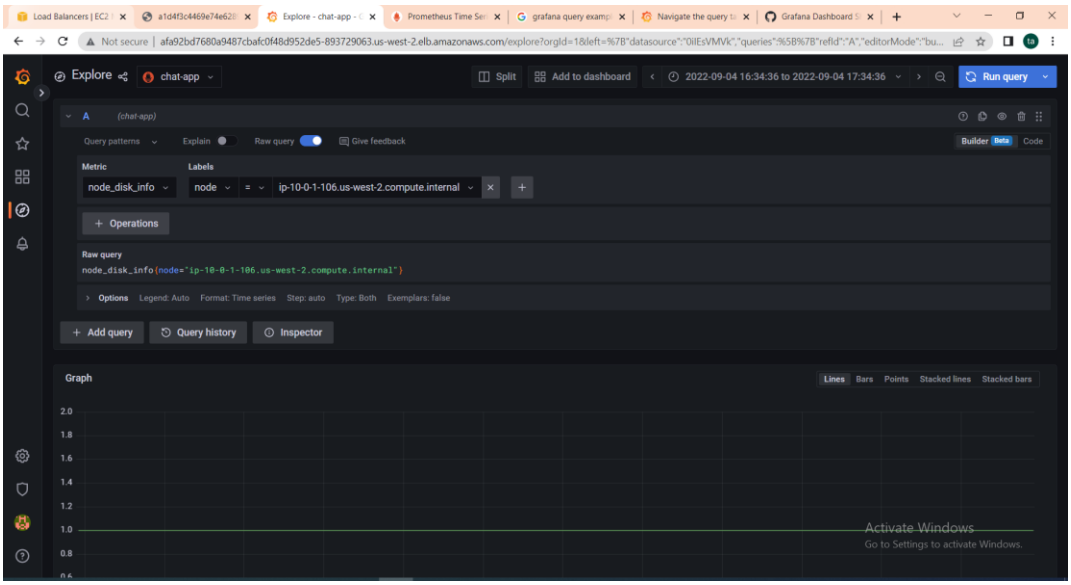
- Network-latency:



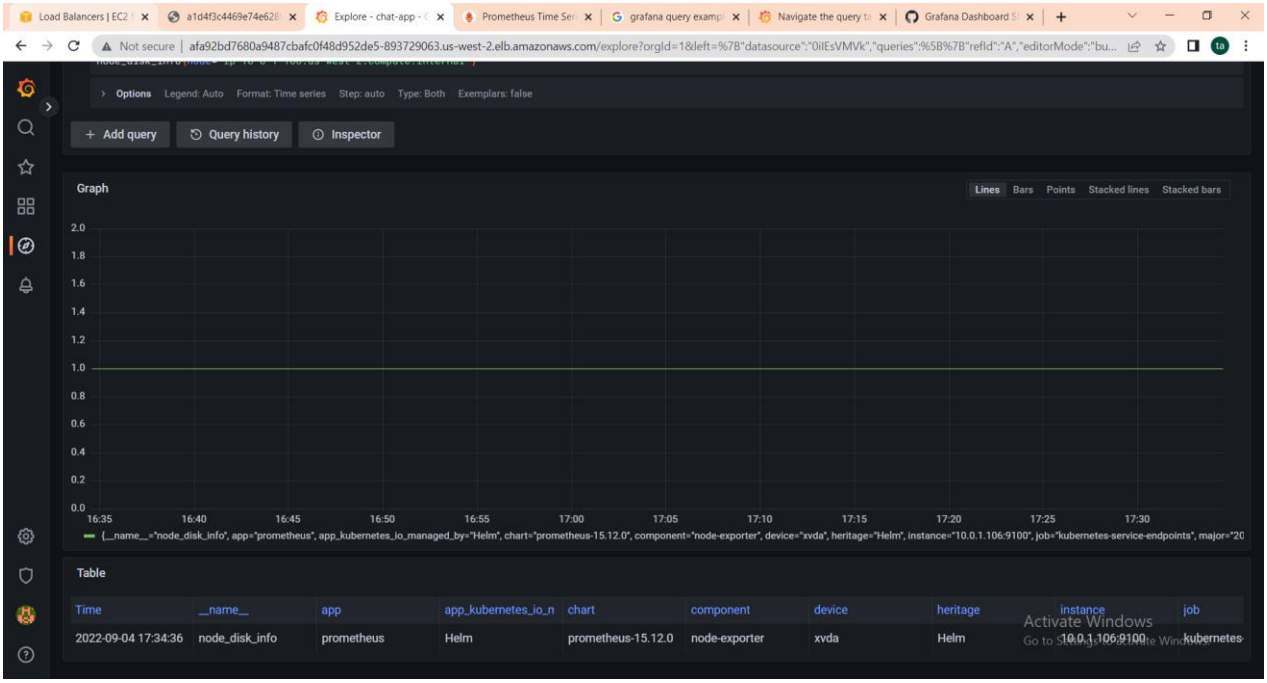
Result:



- Disk utilization:



Result:


Not secure | afa92bd7680a9487cbafc0f48d952de5-893729063.us-west-2.elb.amazonaws.com/explore?orgId=1&left=%7B"datasource":"0iIEsVMVk","queries":%5B%7B"refId":"A","editorMode":"bu...
Options Legend: Auto Format: Time series Step: auto Type: Both Exemplars: false
Add query
Query history
Inspector
Graph
Lines Bars Points Stacked lines Stacked bars
2.0
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
16:35 16:40 16:45 16:50 16:55 17:00 17:05 17:10 17:15 17:20 17:25 17:30
{__name__="node_disk_info", app="prometheus", app_kubernetes_io_managed_by="Helm", chart="prometheus-15.12.0", component="node-exporter", device="xvda", heritage="Helm", instance="10.0.1.106:9100", job="kubernetes-service-endpoints", major="20
Table
Time __name__ app app_kubernetes_io_n chart component device heritage instance job
2022-09-04 17:34:36 node_disk_info prometheus Helm prometheus-15.12.0 node-exporter xvda Helm 10.0.1.106:9100 kubernetes
Activate Windows