**ELK, PROMETHEUS AND GRAFANA**

**Assignments handson cmds:**

1 apt-get update && apt-get install -y apt-transport-https && apt install docker.io -y && systemctl start docker && systemctl enable docker && curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -

2 cat <<EOF >/etc/apt/sources.list.d/kubernetes.list

deb http://apt.kubernetes.io/ kubernetes-xenial main

EOF

3 apt-get update && apt-get install -y kubelet kubeadm kubectl kubernetes-cni

4 cat <<EOF >/etc/docker/daemon.json

{

"exec-opts": ["native.cgroupdriver=systemd"]

}

EOF

5 systemctl restart docker

6 systemctl status docker

7 kubeadm init --ignore-preflight-errors all

8 mkdir -p $HOME/.kube && sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config && sudo chown $(id -u):$(id -g) $HOME/.kube/config

9 kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')"

10 kubectl get pods --all-namespaces

11 kubectl get nodes

12 mkdir -p $HOME/.kube

13 sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

14 sudo chown $(id -u):$(id -g) $HOME/.kube/config

15 kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')"

16 kubectl get pods --all-namespaces

17 kubectl get nodes

18 git clone https://github.com/qf-devops/kubernetes-elk.git

19 ls

20 cd kubernetes-elk/

21 ls

22 cd monitoring/

23 ls -lrt

24 cd kubernetes-elk/

25 ls -lrt

26 cd

27 kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.5.0/aio/deploy/recommended.yaml

28 ls lrt

29 ls -lrt

30 kubectl get svc -n kubernetes-dashboard

31 kubectl edit svc kubernetes-dashboard -n kubernetes-dashboard

32 kubectl get svc -n kubernetes-dashboard

33 vi role.yml

34 kubectl apply -f role.yml

35 clear

36 kubectl -n kubernetes-dashboard get secret $(kubectl -n kubernetes-dashboard get sa/admin-user -o jsonpath="{.secrets[0].name}") -o go-template="{{.data.token | base64decode}}"

37 cd

38 cd kubernetes-elk/

39 cd monitoring/

40 ls

41 cd kubernetes-elk/

42 ls -lrt

43 kubectl apply -f elasticsearch-ss.yaml

44 kubectl get -f elasticsearch-ss.yaml

45 kubectl apply -f logstash-deployment.yaml -f kibana-deployment.yaml

46 kubectl get svc --all-namespaces

47 kubectl apply -f filebeat-ds.yaml

48 cat ~/role.yml

49 ls p-lrt

50 ls -lrt

51 vi filebeat-ds.yaml

52 kubectl apply -f filebeat-ds.yaml

53 kubectl -n kubernetes-dashboard get secret $(kubectl -n kubernetes-dashboard get sa/admin-user -o jsonpath="{.secrets[0].name}") -o go-template="{{.data.token | base64decode}}"

54 vi metricbeat-ds.yaml

55 kubectl apply -f metricbeat-ds.yaml

56 cd ..

57 ls

58 cd monitoring/

59 ls -lrt

60 cd kubernetes-prometheus/

61 ls -lrt

62 kubectl create namespace monitoring

63 kubectl get deployments --namespace=monitoring

64 kubectl get pods --namespace=monitoring

65 cd

66 pwd

67 ll

68 history

69 cd kubernetes-elk/

70 ls -lrt

71 cd monitoring/

72 ls -lrt

73 cd kubernetes-prometheus/

74 ls -lrt

75 kubectl create -f clusterRole.yaml

76 vi clusterRole.yaml

77 kubectl create -f clusterRole.yaml

78 kubectl create namespace monitoring

79 vi config-map.yaml

80 kubectl create -f config-map.yaml

81 kubectl create -f prometheus-deployment.yaml

82 kubectl get deployments --namespace=monitoring

83 kubectl get pods --namespace=monitoring

84 prometheus-deployment-84f65c89c5-mnvkz 8080:9090 -n monitoring

85 prometheus-monitoring-84f65c89c5-mnvkz 8080:9090 -n monitoring

86 kubectl port-forward prometheus-deployment-84f65c89c5-mnvkz 8080:9090 -n monitoring

87 kubectl create -f prometheus-service.yaml --namespace=monitoring

88 kubectl port-forward prometheus-deployment-84f65c89c5-mnvkz 8080:9090 -n monitoring

89 kubectl get pods

90 kubectl get svc -n=monitoring

91 cd ..

92 ls -lrt

93 cd kubernetes-grafana/

94 ls -lrt

95 kubectl create -f grafana-datasource-config.yaml

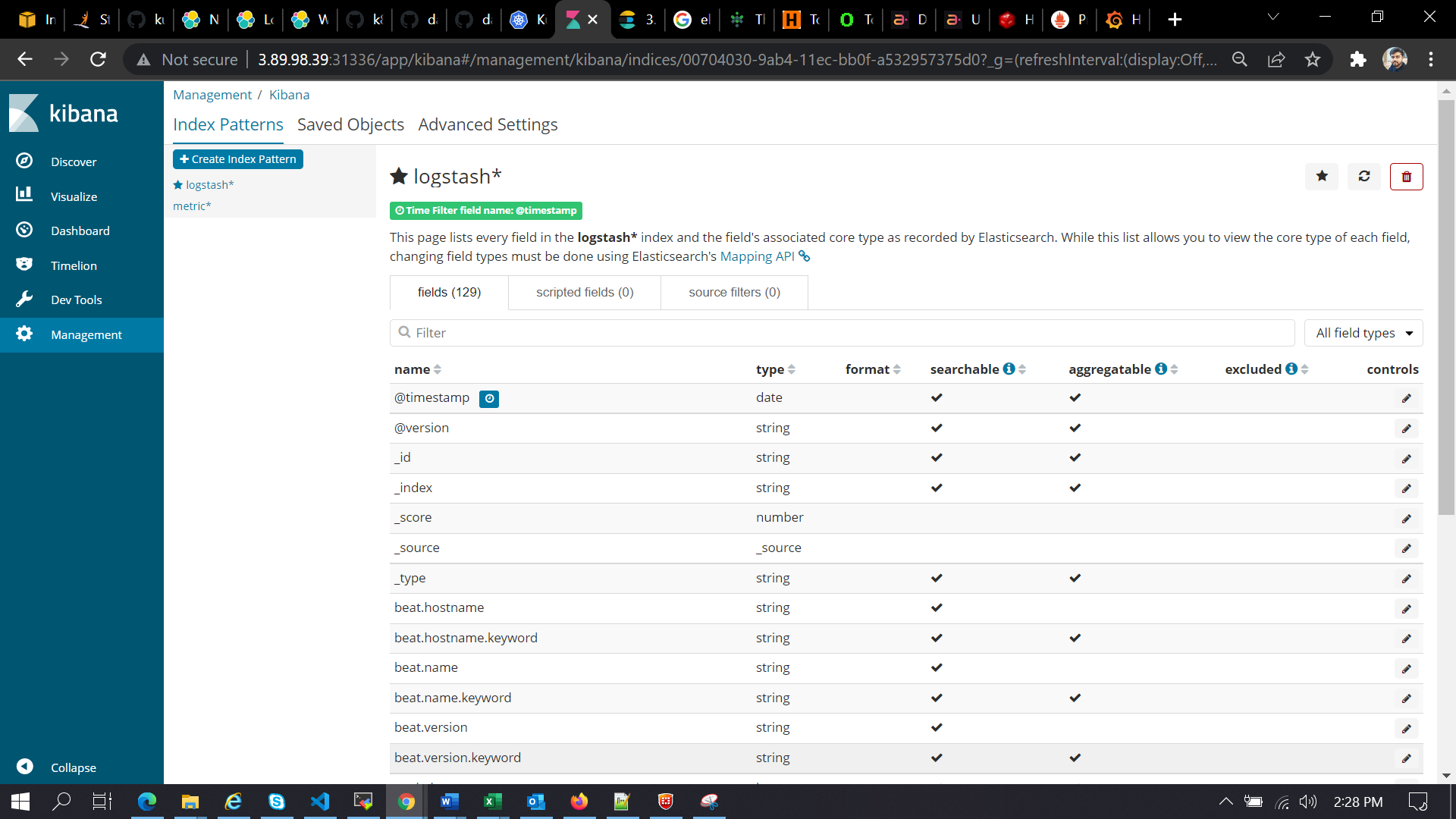
96 kubectl create -f deployment.yaml

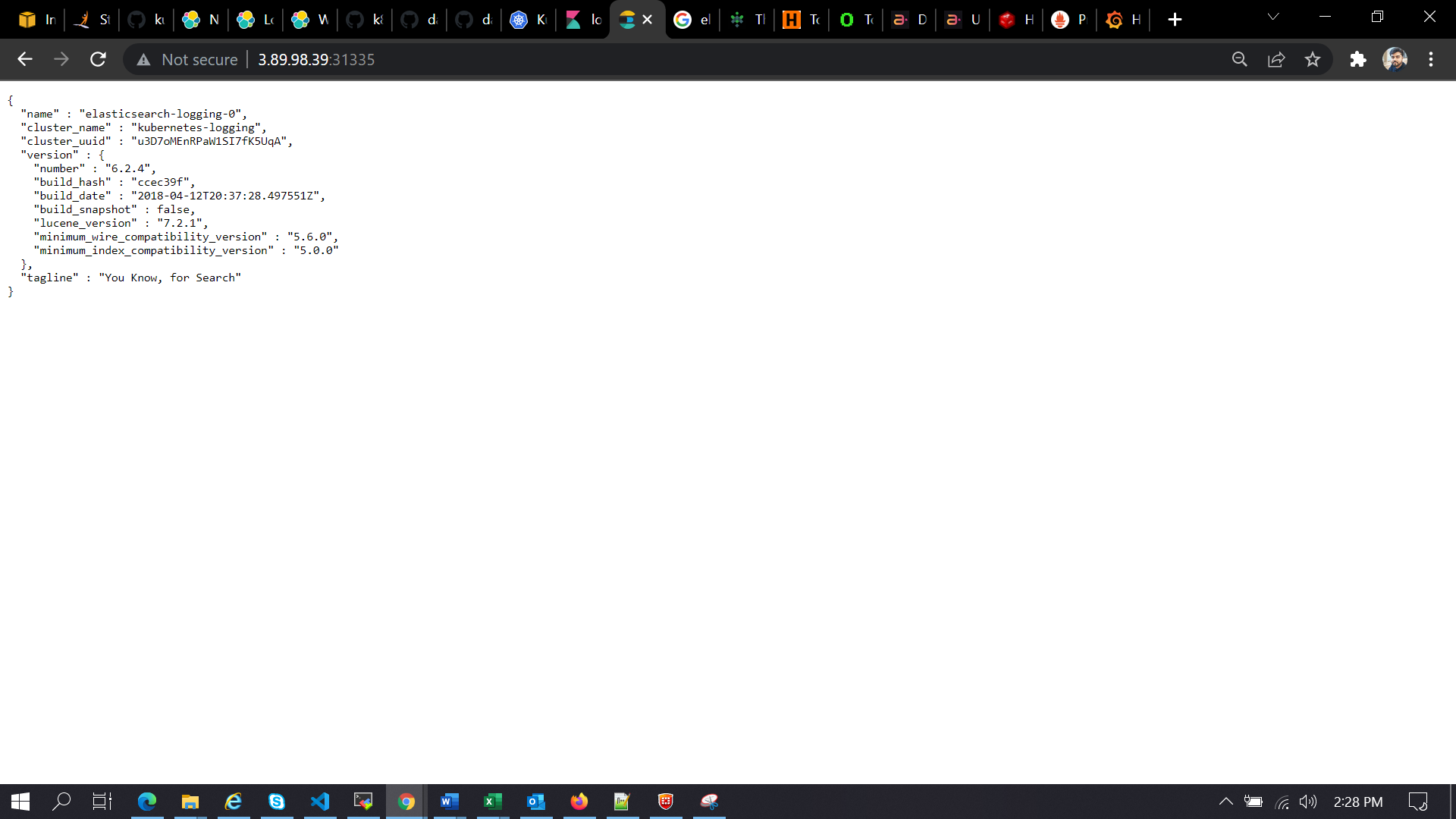
97 kubectl create -f service.yaml

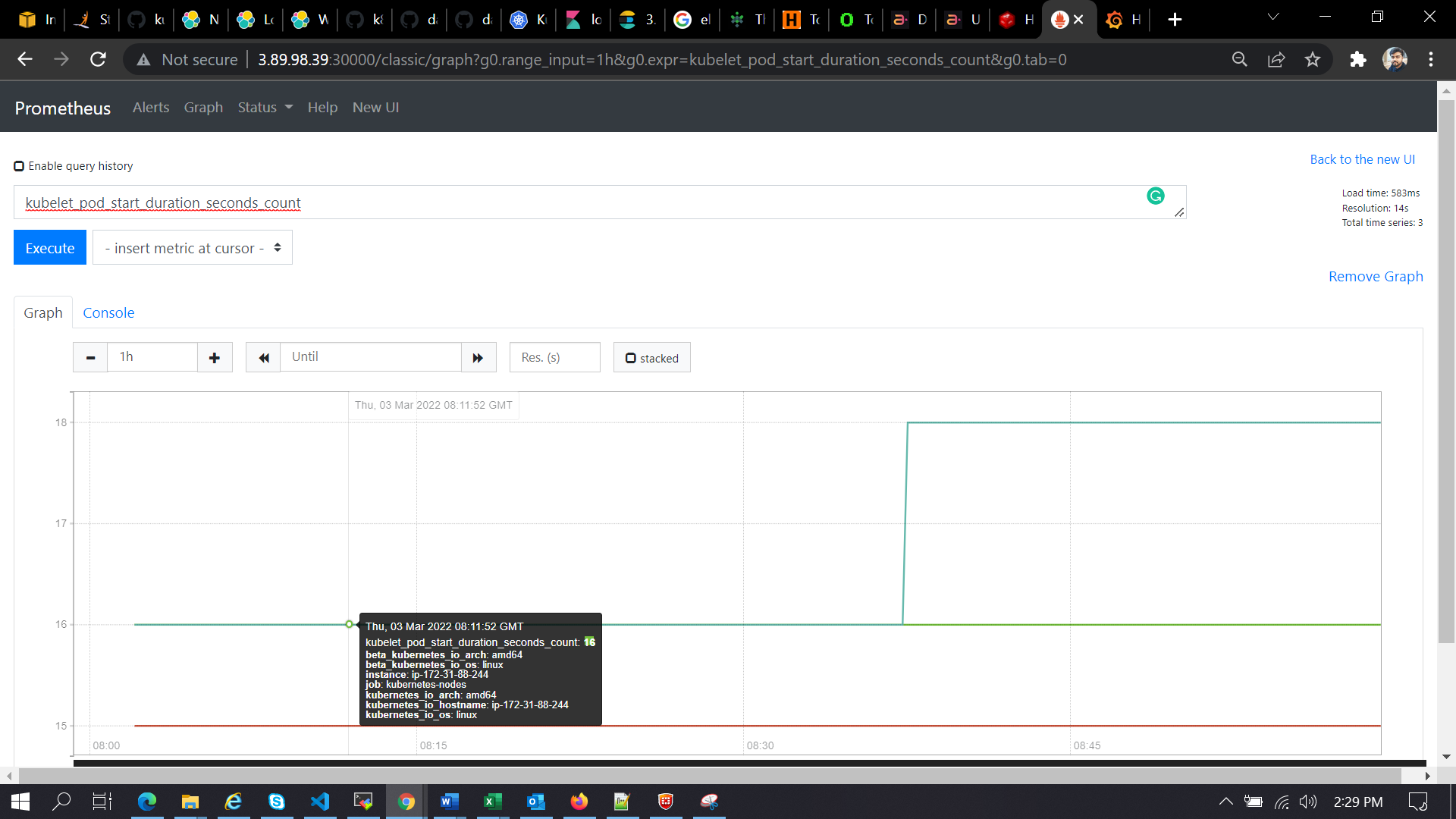
98 kubectl get svc

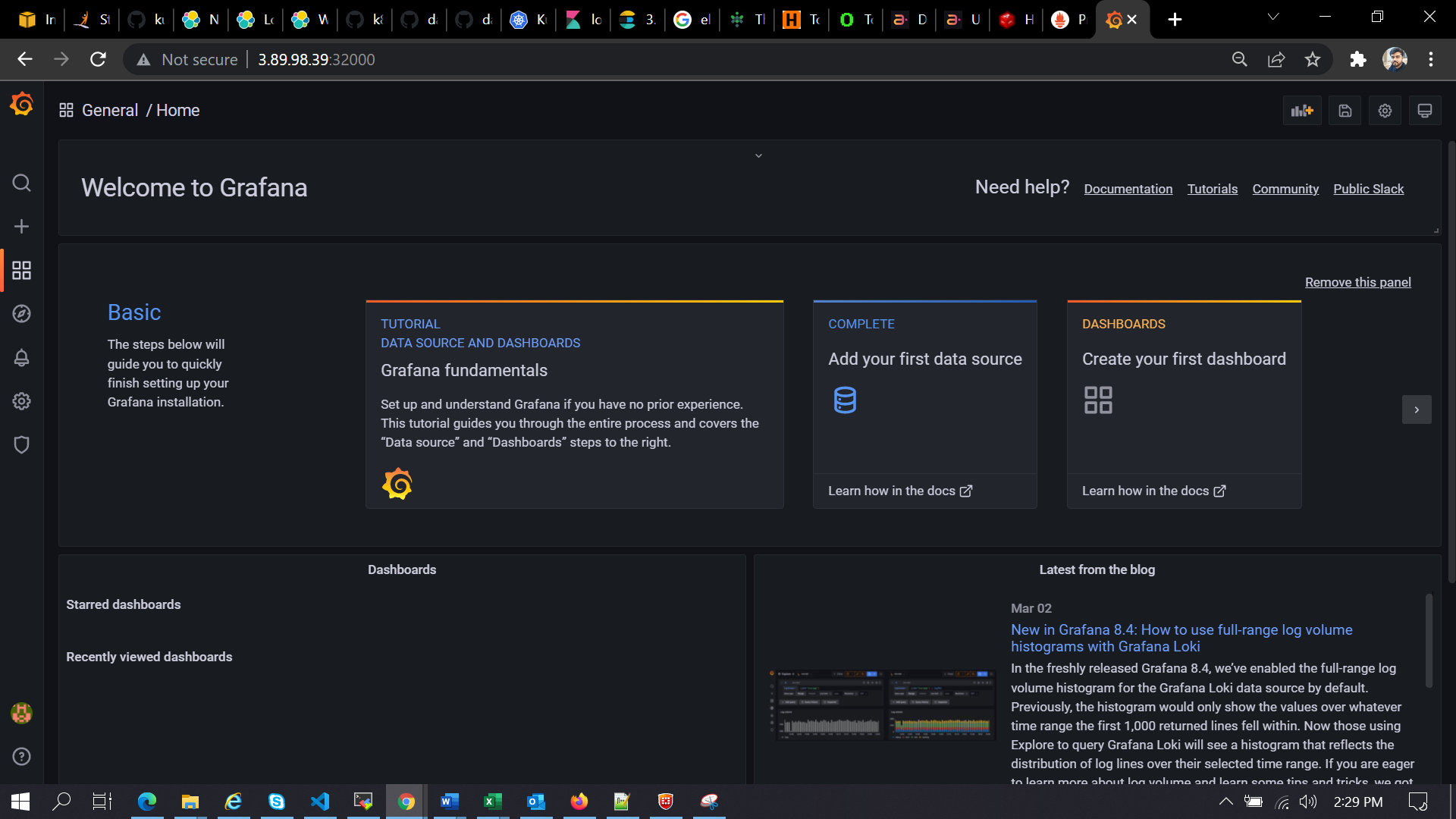
99 kubectl get svc -n=monitoring

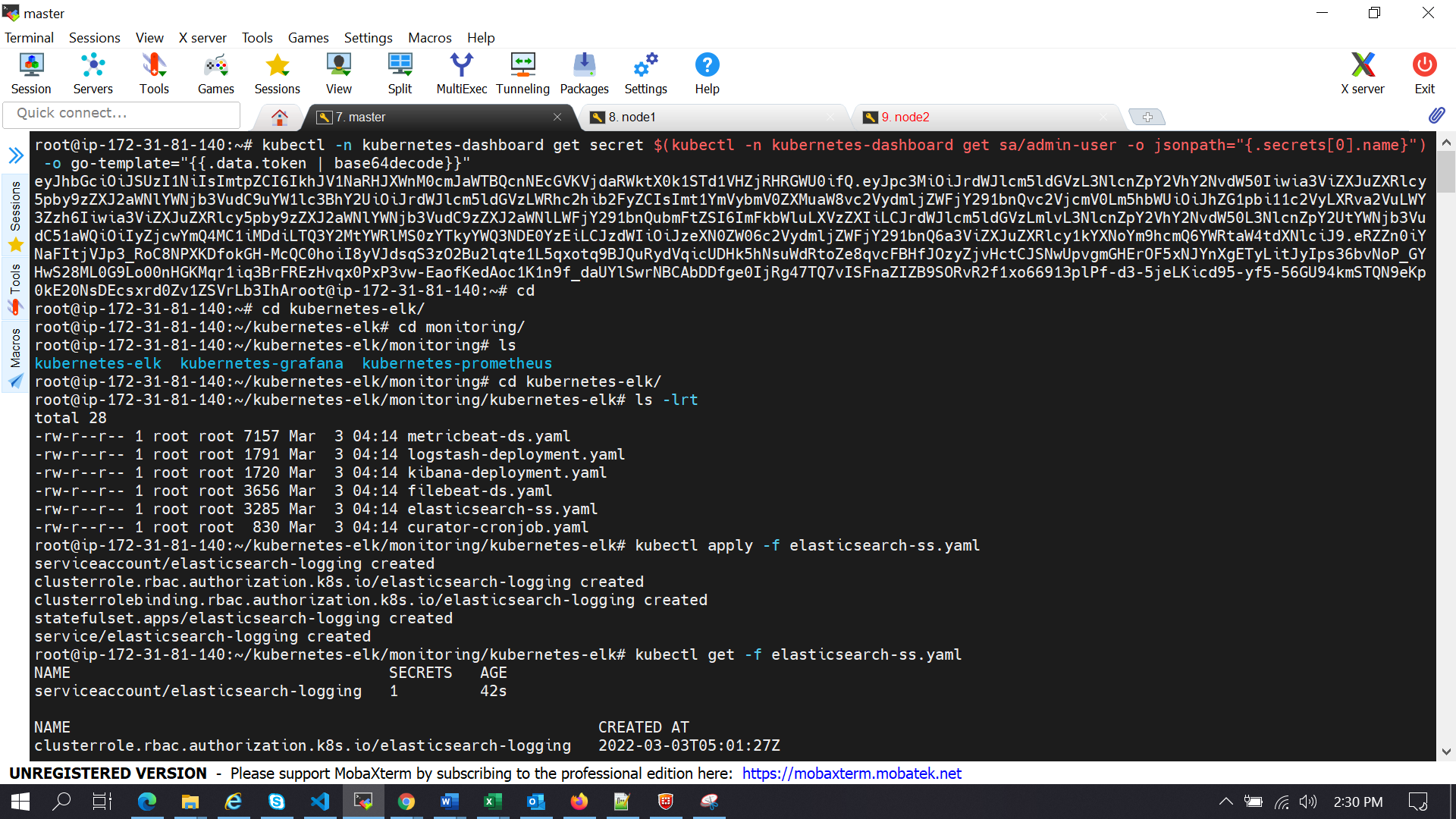
**Screenshots:**

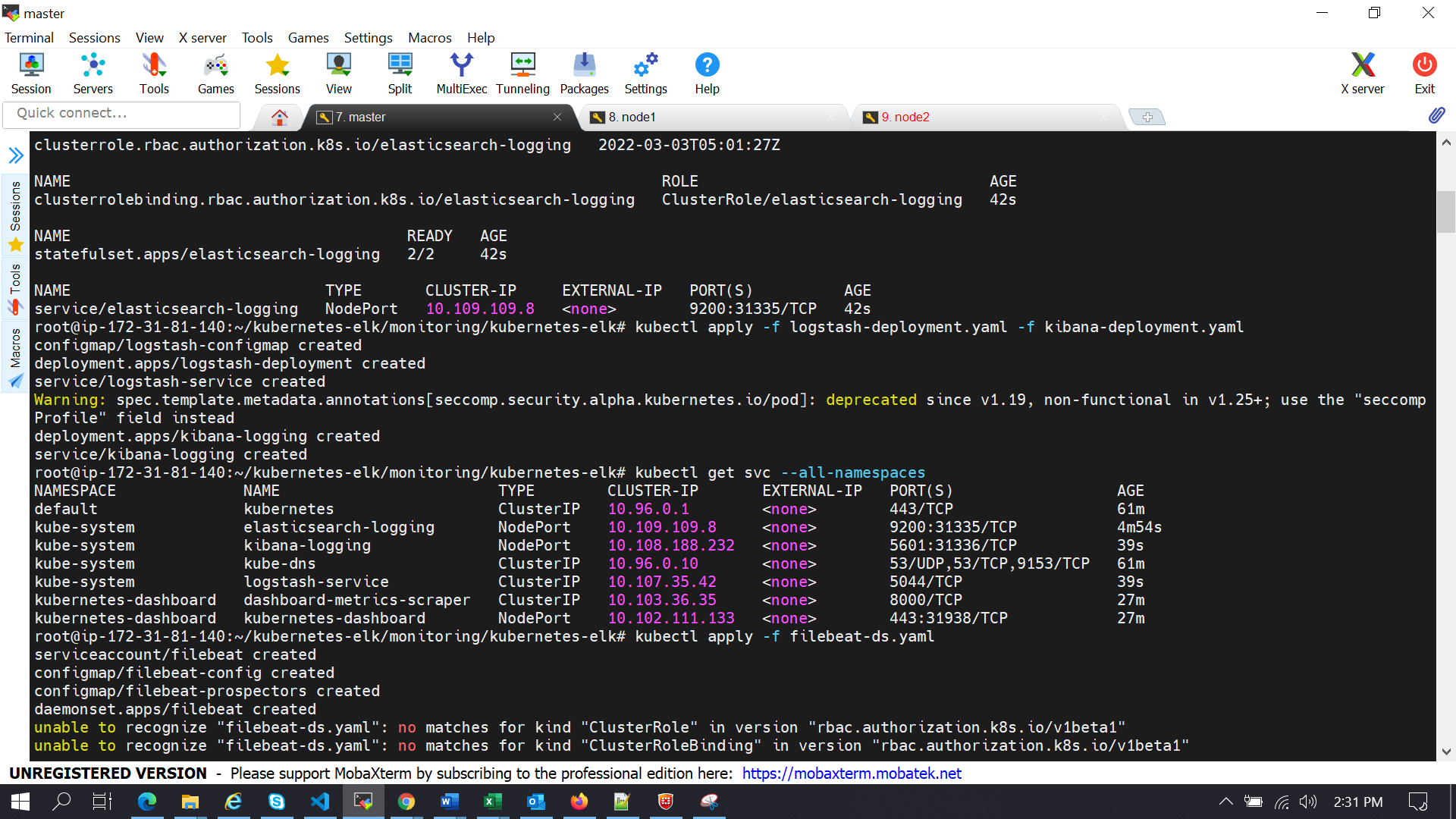


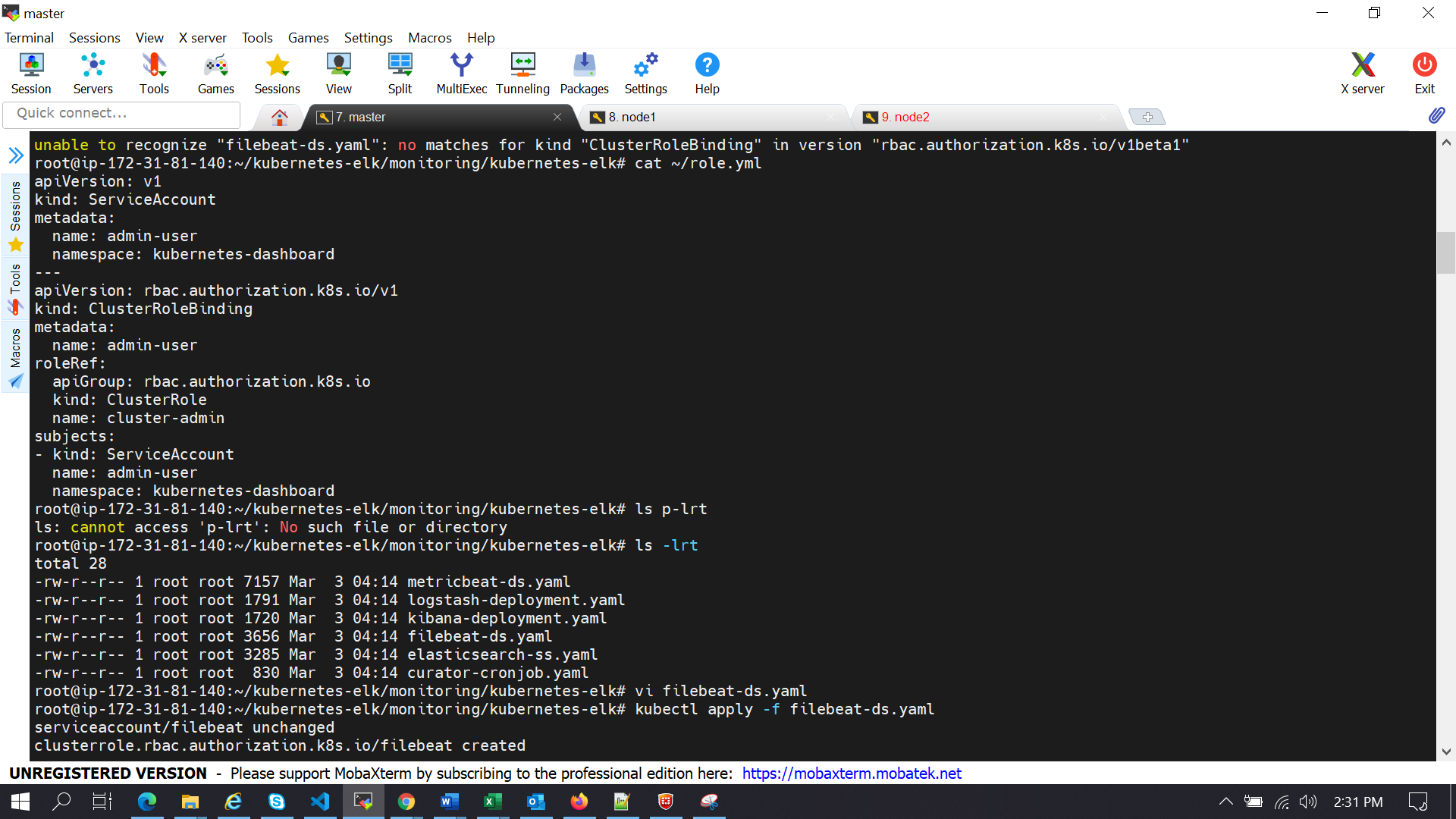


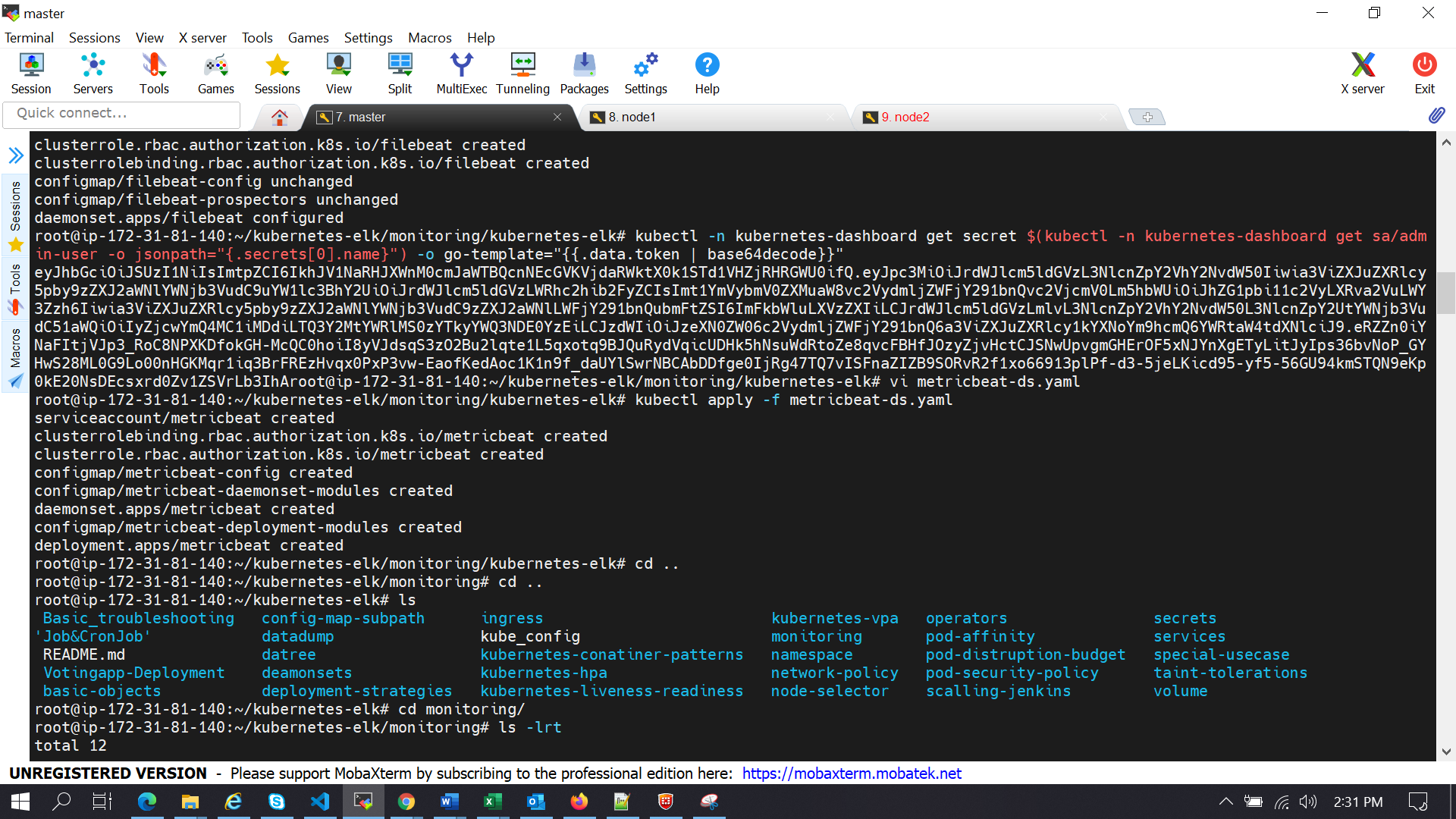


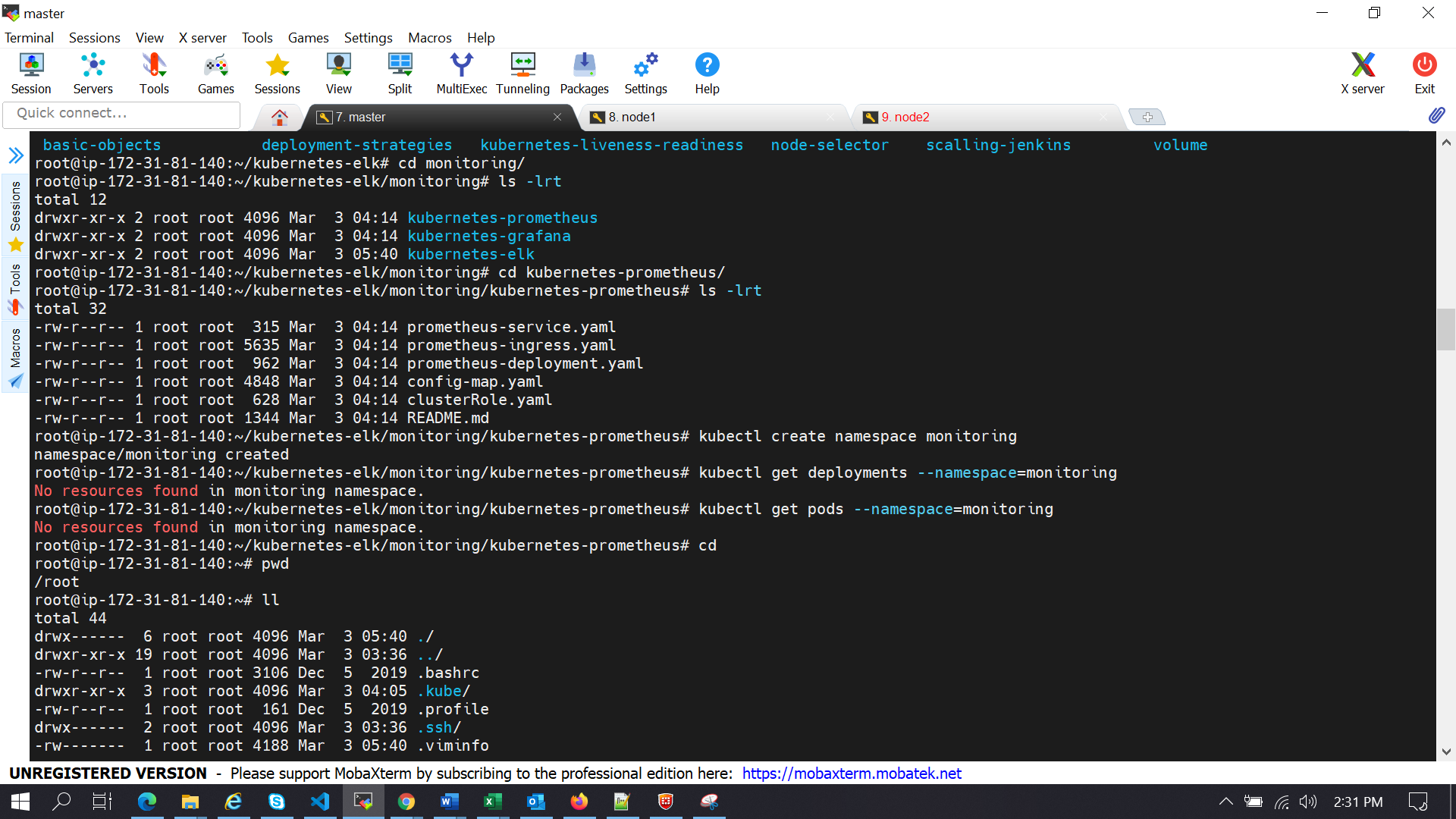


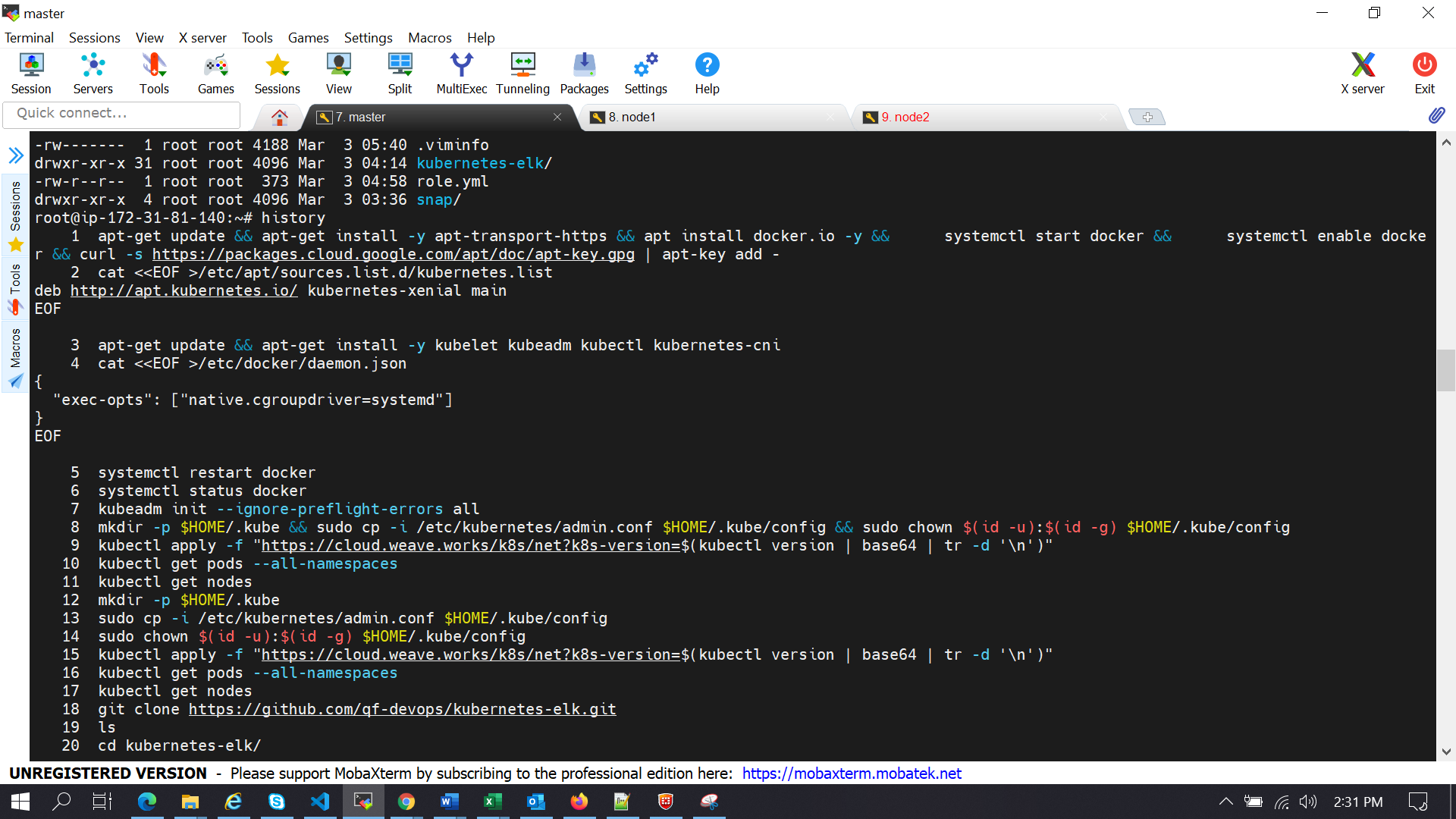


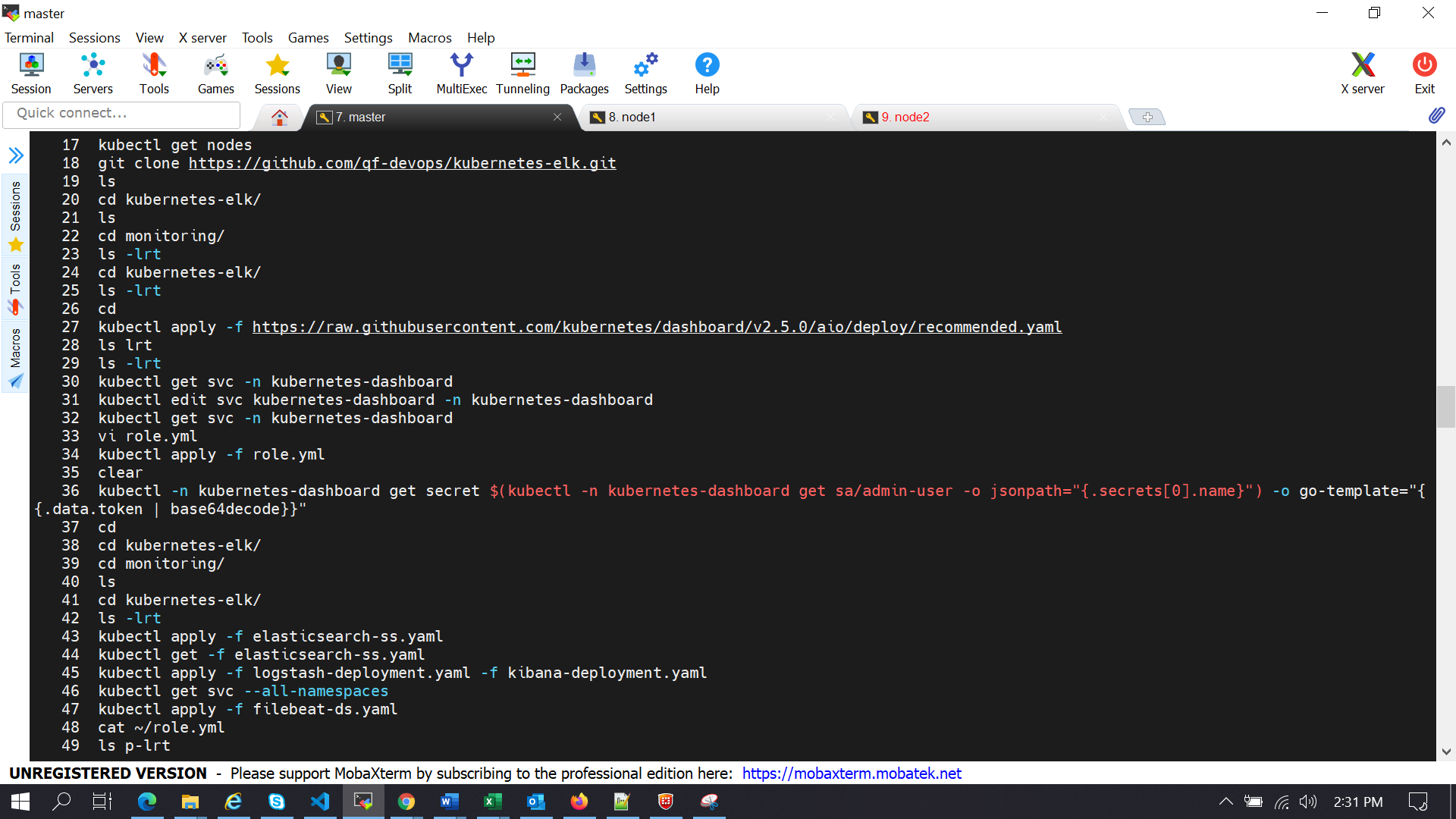


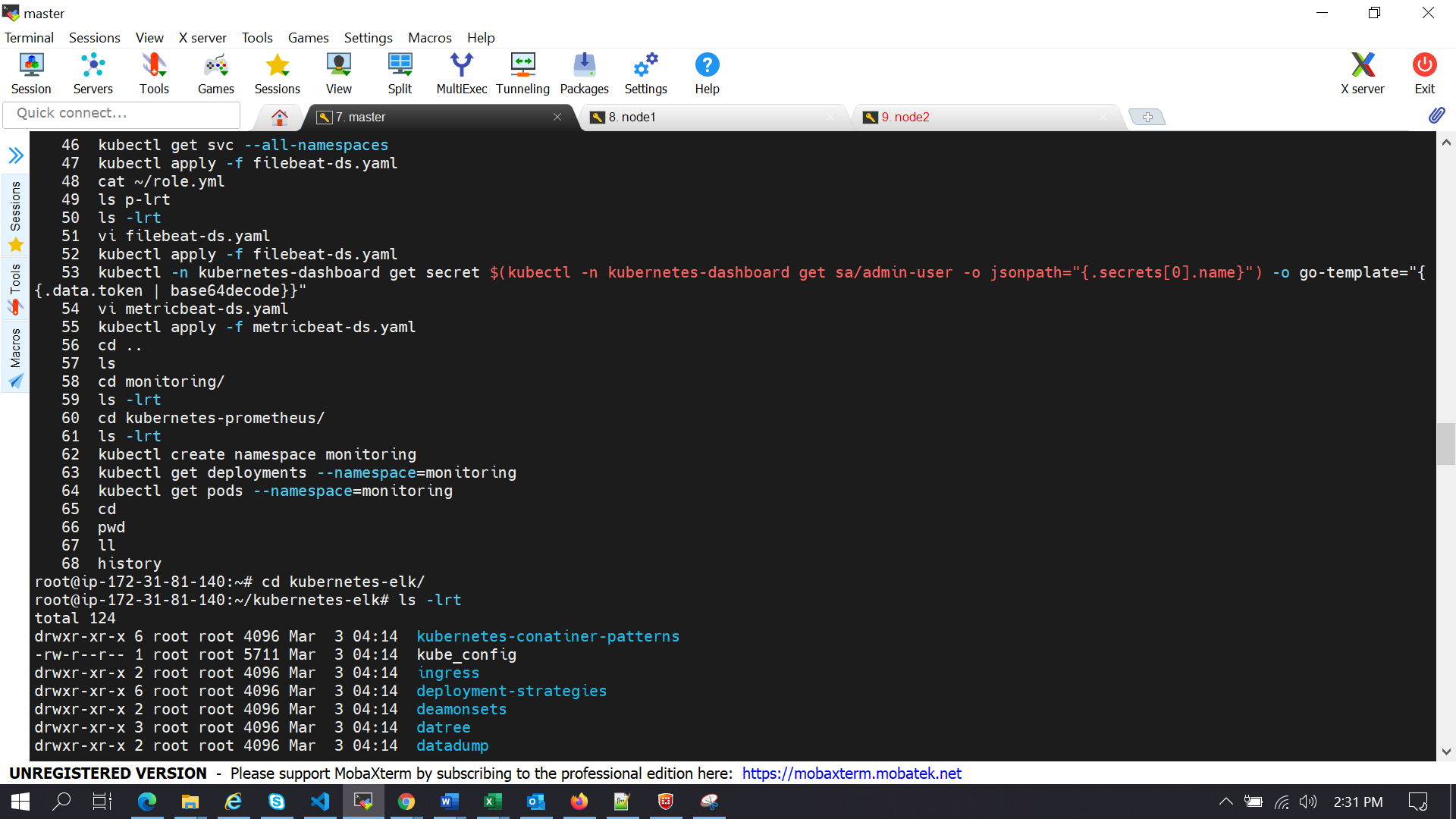


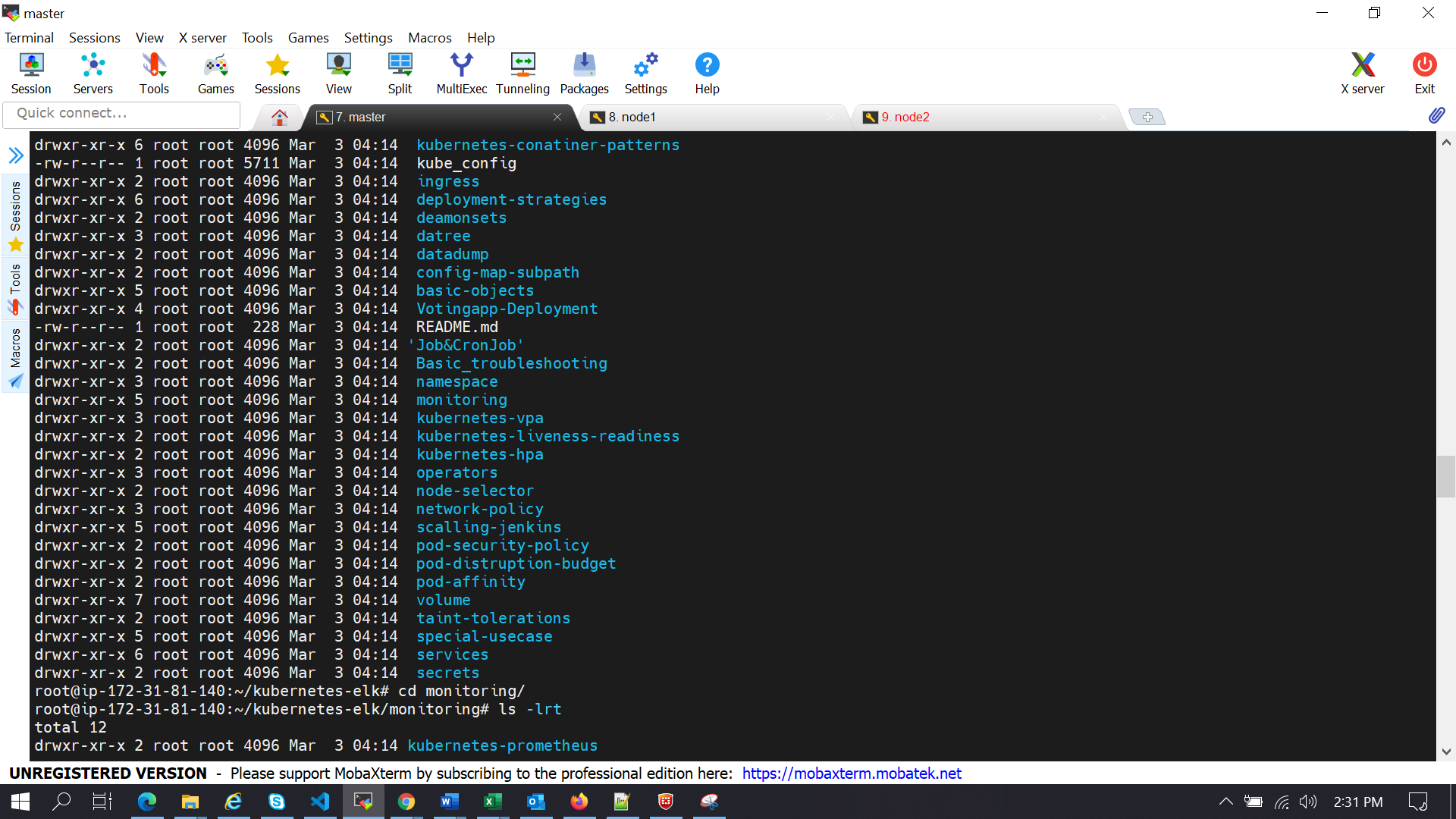


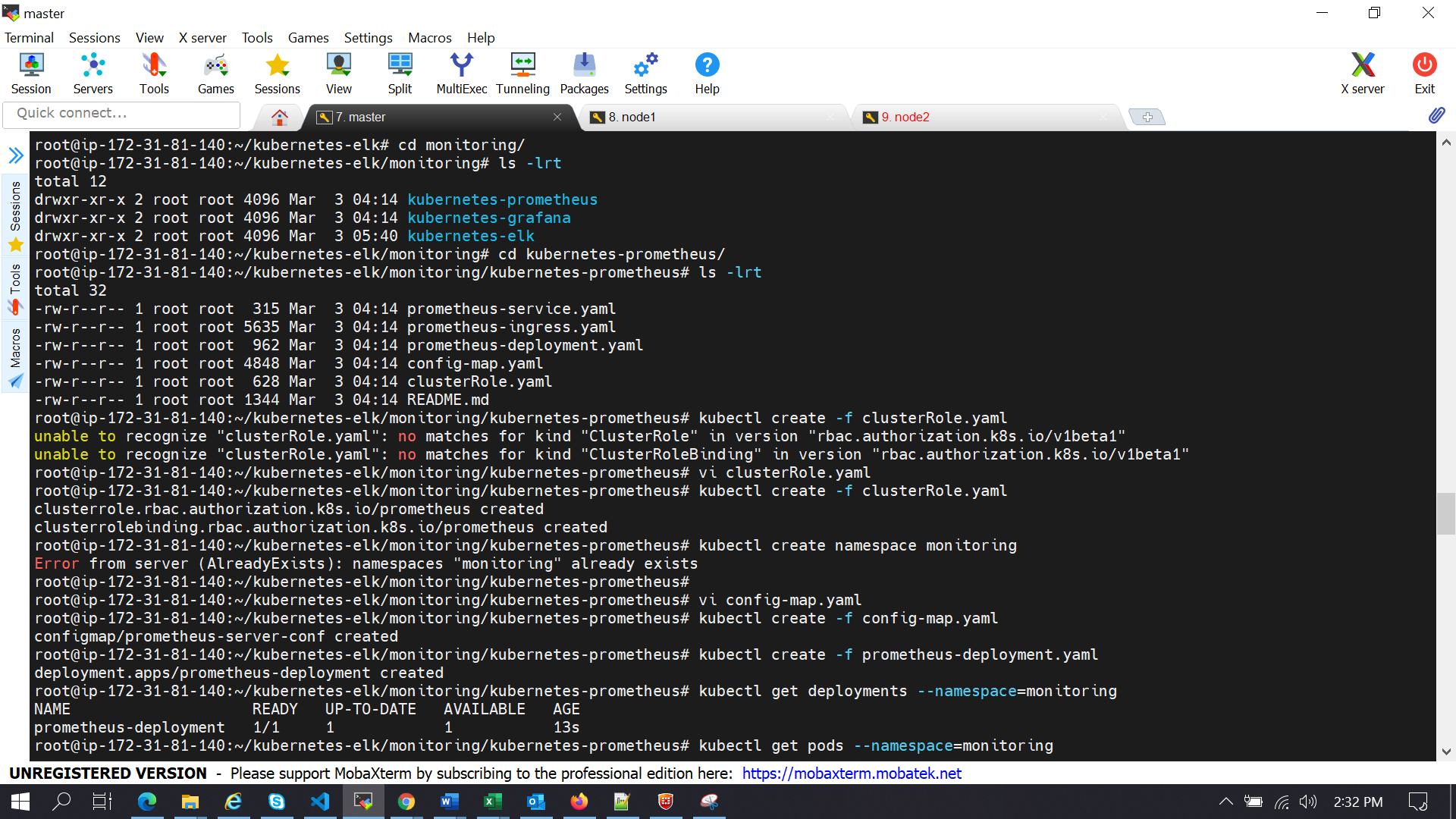


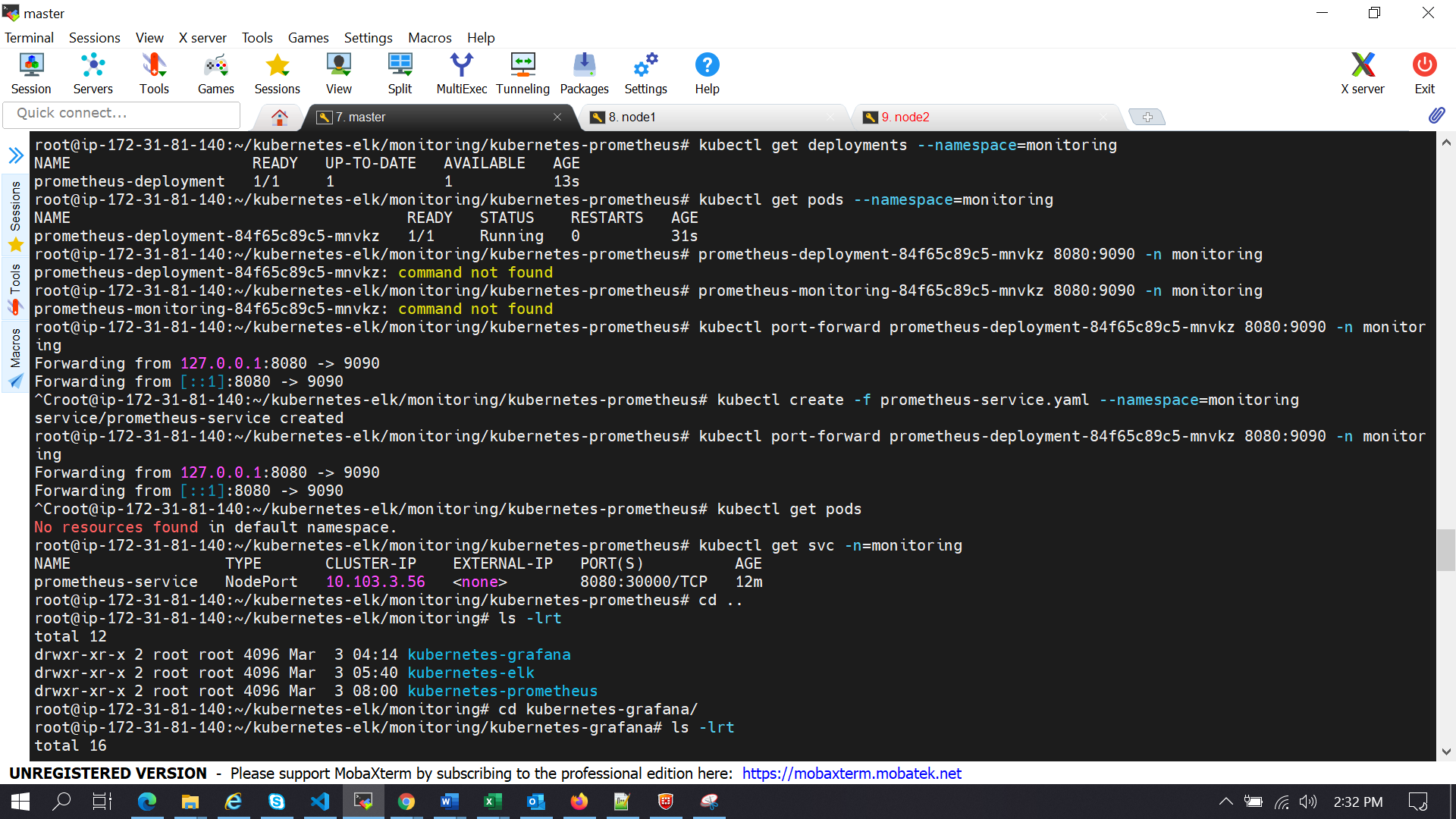


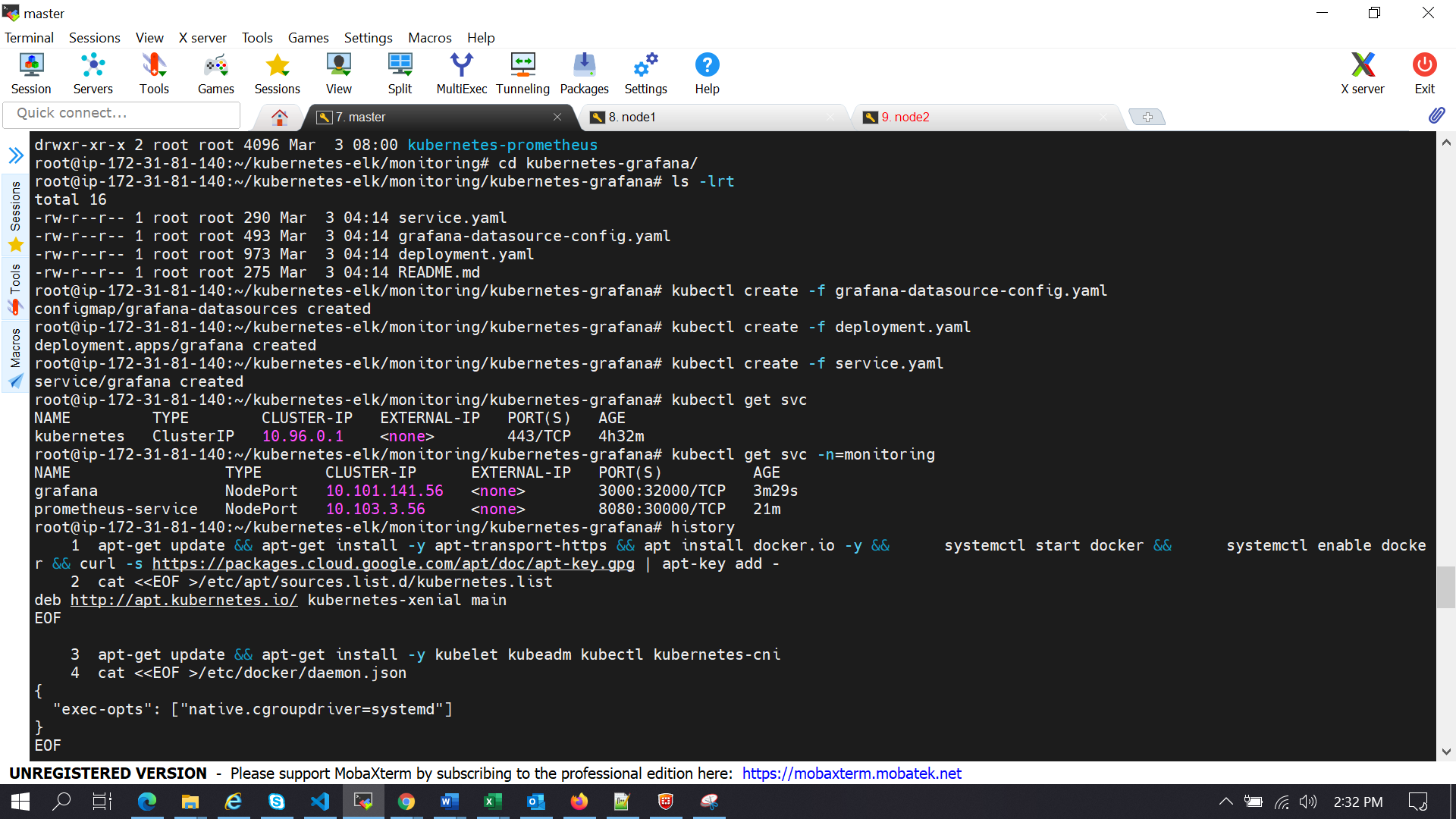












**For Nodes:**

