**04. Review Kubernetes Manifests & Node Group Role**

--- Reference - <https://github.com/stacksimplify/aws-eks-kubernetes-masterclass/blob/master/10-ECR-Elastic-Container-Registry-and-EKS/01-aws-ecr-kubenginx/index.html>

**Using ECR Image with Amazon EKS**

--- Review the k8s manifests

--- Understand the Deployment and Service kubernetes manifests present in folder 10-ECR-Elastic-Container-Registry\02-kube-manifests

1. Deployment: 01-ECR-Nginx-Deployment.yml
2. NodePort Service: 02-ECR-Nginx-NodePortService.yml
3. ALB Ingress Service: 03-ECR-Nginx-ALB-IngressService.yml

--- **01-ECR-Nginx-Deployment.yml**

apiVersion: apps/v1

kind: Deployment

metadata:

  name: kubeapp-ecr

  labels:

    app: kubeapp-ecr

spec:

  replicas: 2

  selector:

    matchLabels:

      app: kubeapp-ecr

  template:

    metadata:

      labels:

        app: kubeapp-ecr

    spec:

      containers:

        - name: kubeapp-ecr

          image: 180789647333.dkr.ecr.us-east-1.amazonaws.com/aws-ecr-kubenginx:1.0.0 # image from the AWS ECR.

          resources:

            requests:

              memory: "128Mi"

              cpu: "500m"

            limits:

              memory: "256Mi"

              cpu: "1000m"

          ports:

            - containerPort: 80

--- **02-ECR-Nginx-NodePortService.yml**

apiVersion: v1

kind: Service

metadata:

  name: kubeapp-ecr-nodeport-service

  labels:

    app: kubeapp-ecr

  annotations:

#Important Note:  Need to add health check path annotations in service level if we are planning to use multiple targets in a load balancer

    alb.ingress.kubernetes.io/healthcheck-path: /index.html

spec:

  type: NodePort

  selector:

    app: kubeapp-ecr

  ports:

    - port: 80

      targetPort: 80

--- **03-ECR-Nginx-ALB-IngressService.yml**

# Annotations Reference:  https://kubernetes-sigs.github.io/aws-alb-ingress-controller/guide/ingress/annotation/

apiVersion: extensions/v1beta1

kind: Ingress

metadata:

  name: ecr-ingress-service

  labels:

    app: kubeapp-ecr

  annotations:

    # Ingress Core Settings

    kubernetes.io/ingress.class: "alb"

    alb.ingress.kubernetes.io/scheme: internet-facing

    # Health Check Settings

    alb.ingress.kubernetes.io/healthcheck-protocol: HTTP

    alb.ingress.kubernetes.io/healthcheck-port: traffic-port

    alb.ingress.kubernetes.io/healthcheck-interval-seconds: '15'

    alb.ingress.kubernetes.io/healthcheck-timeout-seconds: '5'

    alb.ingress.kubernetes.io/success-codes: '200'

    alb.ingress.kubernetes.io/healthy-threshold-count: '2'

    alb.ingress.kubernetes.io/unhealthy-threshold-count: '2'

    ## SSL Settings

    alb.ingress.kubernetes.io/listen-ports: '[{"HTTPS":443}, {"HTTP":80}]'

    alb.ingress.kubernetes.io/certificate-arn: arn:aws:acm:us-east-1:180789647333:certificate/9f042b5d-86fd-4fad-96d0-c81c5abc71e1

    #alb.ingress.kubernetes.io/ssl-policy: ELBSecurityPolicy-TLS-1-1-2017-01 #Optional (Picks default if not used)

    # SSL Redirect Setting

    alb.ingress.kubernetes.io/actions.ssl-redirect: '{"Type": "redirect", "RedirectConfig": { "Protocol": "HTTPS", "Port": "443", "StatusCode": "HTTP\_301"}}'

    # External DNS - For creating a Record Set in Route53

    external-dns.alpha.kubernetes.io/hostname: ecrdemo.kubeoncloud.com

spec:

  rules:

    - http:

        paths:

          - path: /\* # SSL Redirect Setting

            backend:

              serviceName: ssl-redirect

              servicePort: use-annotation

          - path: /\*

            backend:

              serviceName: kubeapp-ecr-nodeport-service

              servicePort: 80

# Important Note-1: In path based routing order is very important, if we are going to use  "/\*", try to use it at the end of all rules.

**Verify ECR Access to EKS Worker Nodes**

--- Go to Services -> EC2 -> Running Instances > Select a Worker Node -> Description Tab

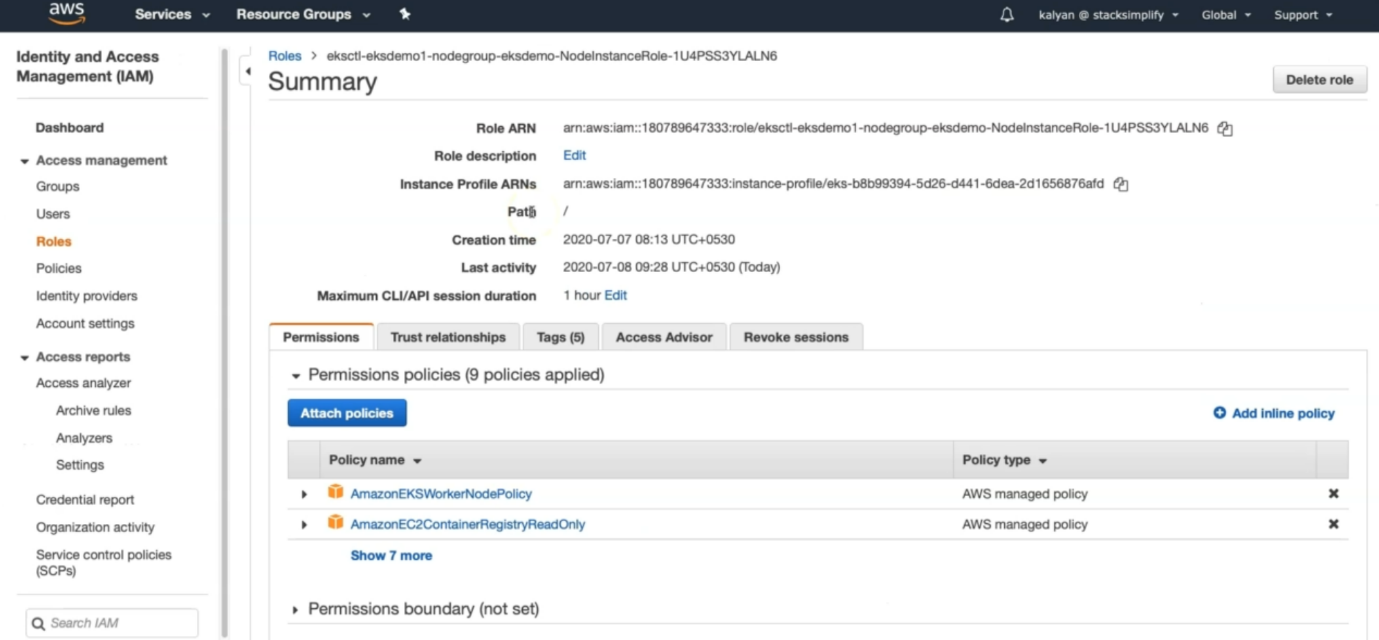
--- Click on value in IAM Role field

**# Sample Role Name**

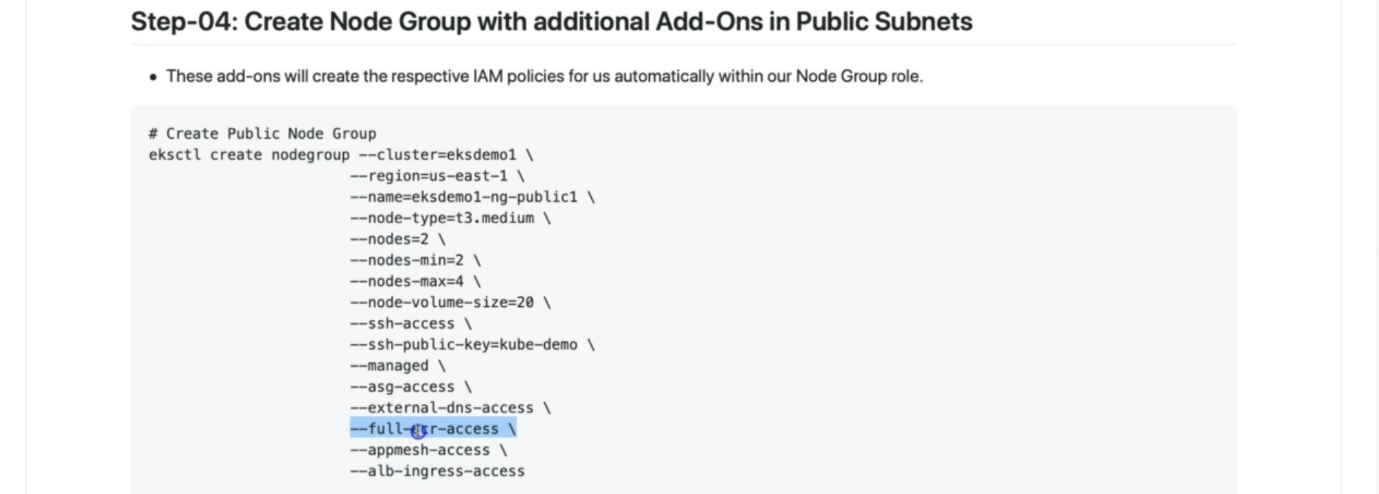
--- eksctl-eksdemo1-nodegroup-eksdemo-NodeInstanceRole-1U4PSS3YLALN6

--- In IAM on that specific role, verify permissions tab

--- policy with name AmazonEC2ContainerRegistryReadOnly, AmazonEC2ContainerRegistryPowerUser should be associated



--- how this policy is attached…? While we creating nodegroup we attached the policy. Look at the command.



--- **note** – in our next lecture, we are going to deploy kubernetes manifests.