Cert Manager Setup

Prerequisites

- Install Helm version 3 or later

Steps

- Add Cert Manager Helm chart.

  `helm repo add jetstack https://charts.jetstack.io`

- Update helm chart repo cache.

  `helm repo update`

- Install Cert-Manager helm chart.

  ```
  cert-manager jetstack/cert-manager \
  --namespace cert-manager \
  --create-namespace \
  --version v1.6.0 \
  --set installCRDs=true
  ```

Setup *ClusterIssuer* to generate Certificate.

- Setting up ***ClusterIssuer*** for same AWS account where cluster is hosted and route53 hosted zone is configured.

  ClusterIssuer-same-account.yaml

```
apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
#name: Name for clusterissuer
name:
spec:
acme:
    #acme.email: Email Address of the AWS Account user.
    email:
    preferredChain: ""
    #acme.privateKeySecretRef.name: Will be used in Ingress
Configuration to point the certificate.
    privateKeySecretRef:
      name:
    #acme.server: Let's Encrypt Production API Server.
    server: https://acme-v02.api.letsencrypt.org/directory
    solvers:
    #acme.solvers.http01: Used for non wildcard URLs.
    - http01:
        ingress:
          class: nginx
    #acme.solvers.dns01: Used for Wildcard URLs.
    - dns01:
        route53:
```

```
     #hostedZoneID: Could be found at AWS Console > Route53 >
Hosted Zone > Select any existing > Hosted Zone Details
     hostedZoneID:
     #acme.dns01.route53.region: Region for AWS Account.
     region:
     secretAccessKeySecretRef:
      name: ""
   #acme.selector.dnsZones: domain name (ex: example.com)
   selector:
    dnsZones:
    - example.com
```

- Setting up ***ClusterIssuer*** for Where cluster is located in different account and route53 hosted zone is configured in different account.

  ClusterIssuer-different-account.yaml

```

apiVersion: cert-manager.io/v1
kind: ClusterIssuer
metadata:
#name: Name for clusterissuer
name:
spec:
acme:
  #acme.email: Email Address of the AWS Account user.
  email:
  preferredChain: ""
  #acme.privateKeySecretRef.name: Will be used in Ingress
Configuration to point the certificate.
  privateKeySecretRef:
   name:
  #acme.server: Let's Encrypt Production API Server.
  server: https://acme-v02.api.letsencrypt.org/directory
  solvers:
  #acme.solvers.http01: Used for non wildcard URLs.
  - http01:
    ingress:
     class: nginx
  #acme.solvers.dns01: Used for Wildcard URLs.
  - dns01:
    route53:
     #acme.solvers.dns01.route53.hostedZoneID: Could be found at
AWS Console > Route53 > Hosted Zone > Select any existing > Hosted
Zone Details
     hostedZoneID:
     #acme.solvers.dns01.route53.accessKeyID: AWS SDK Access Key
     accessKeyID:
     #acme.solvers.dns01.route53.region: Region for AWS Account.
     region:
     secretAccessKeySecretRef:

#acme.solvers.dns01.route53.secretAccessKeySecretRef.name: Secret
name in Kubernetes cluster where AWS SDK Secret API Key is stored.
      name:
            #acme.solvers.dns01.route53.secretAccessKeySecretRef.key: Inside of Kubernetes Secret's key pair. ex: aws-secret-key: 294snnsicsdbnwiw9du7274h
            key:

      #acme.solvers.selector.dnsZones: domain name (ex: example.com)
      selector:
        dnsZones:
        - example.com
```

- Setting up ***Ingress*** Configuration.

  ingress.yaml

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  annotations:
    cert-manager.io/cluster-issuer: cluster-controlcenter # Secret name, Value of acme.privateKeySecretRef.name in ClusterIssuer YAML Configuration file
    kubernetes.io/ingress.class: nginx
    nginx.ingress.kubernetes.io/proxy-connect-timeout: "1000"
    nginx.ingress.kubernetes.io/proxy-read-timeout: "1000"
    nginx.ingress.kubernetes.io/proxy-send-timeout: "1000"
  labels:
    app: controlcenter
    env: Production
  name: ingress-controlcenter
  namespace: ns-prod
spec:
  rules:
  - host: '*.example.com'
    http:
      paths:
      - backend:
          serviceName: svc-controlcenter
          servicePort: 80
        path: /
        pathType: Prefix
  tls:
  - hosts:
    - '*.example.com'
    secretName: cluster-controlcenter # Secret name, Value of acme.privateKeySecretRef.name in ClusterIssuer YAML Configuration file
```