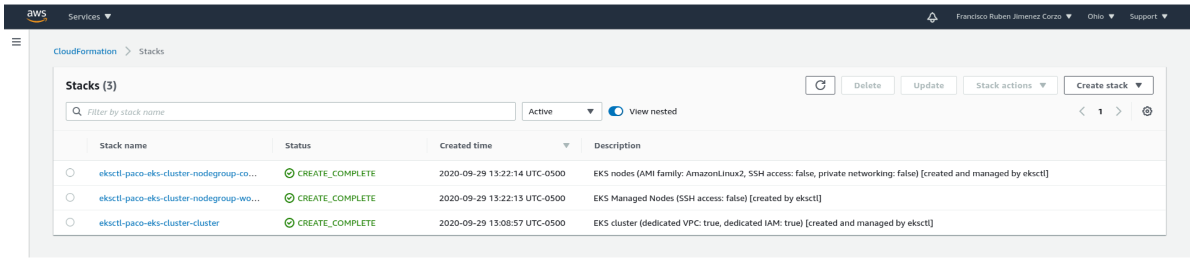
AWS-EKS-Cert-Manager Example

Descripcion General del Caso de Uso:

Se desea crear un sitio tipo tienda en linea, denominado: https://my-store-demo.click/, a manera de ejemplo de lo que se puede hacer en un Cluster de Kubernetes, para este caso de uso, empleando Amazon Elastic Kubernetes Service (**EKS**), asi mismo, empleando un controlador Nginx para permitir la correcta operacion de un Ingress para controlar los flujos de red del sitio dependiendo del path de navegacion. Por ultimo, el sitio deberá contar con un certificado valido TLS. Todo lo anterior, minimizando los costos de implementacion.

Descripcion detallada de la implementacion del caso de uso:

1. Crear un cluster de Kubernetes
   (1) Elegi crearlo Managed, **EKS**: Amazon Elastic Kubernetes Service
   (2) Emplear la linea de comando en lugar de la interfaz Web, con eksctl version 0.28.1
   (3) Genere un YAML, llamado eks-cluster.yaml
   (4) Se creo con el comando:
      - eksctl create cluster -f eks-cluster.yaml
        Este comando en realidad emplea CloudFormation stacks para crear el cluster, detalle:

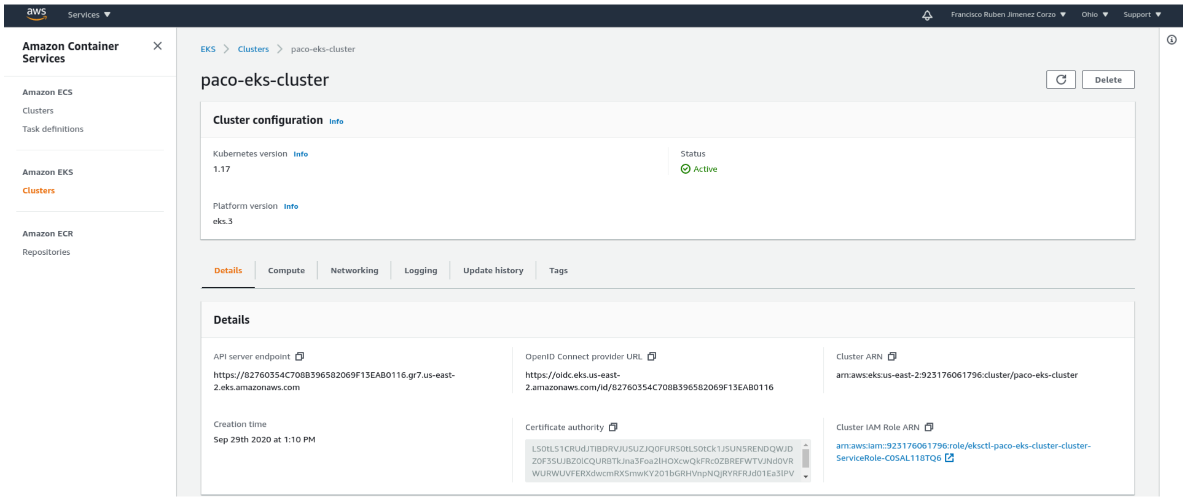


	Stack name	Status	Created time	Description
○	eksctl-paco-eks-cluster-nodegroup-co...	✓ CREATE_COMPLETE	2020-09-29 13:22:14 UTC-0500	EKS nodes (AMI family: AmazonLinux2, SSH access: false, private networking: false) [created and managed by eksctl]
○	eksctl-paco-eks-cluster-nodegroup-wo...	✓ CREATE_COMPLETE	2020-09-29 13:22:13 UTC-0500	EKS Managed Nodes (SSH access: false) [created by eksctl]
○	eksctl-paco-eks-cluster-cluster	✓ CREATE_COMPLETE	2020-09-29 13:08:57 UTC-0500	EKS cluster (dedicated VPC: true, dedicated IAM: true) [created and managed by eksctl]

   (5) Probar el cluster:

```
$ kubectl cluster-info

kubectl get Kubernetes master is running at https://82760354C708B396582069F13EAB0116.gr7.us-east-
2.eks.amazonaws.com

CoreDNS is running at https://82760354C708B396582069F13EAB0116.gr7.us-east-
2.eks.amazonaws.com/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.

$ kubectl get nodes -o wide

NAME                                  STATUS   ROLES    AGE    VERSION             INTERNAL-IP
EXTERNAL-IP     OS-IMAGE      KERNEL-VERSION               CONTAINER-RUNTIME

ip-192-168-28-169.us-east-2.compute.internal   Ready    <none>   3h49m   v1.17.11-eks-cfdc40
192.168.28.169   3.129.43.230    Amazon Linux 2   4.14.193-149.317.amzn2.x86_64   docker://19.3.6

ip-192-168-90-145.us-east-2.compute.internal   Ready    <none>   3h48m   v1.17.11-eks-cfdc40
192.168.90.145   3.135.216.178   Amazon Linux 2   4.14.193-149.317.amzn2.x86_64   docker://19.3.6

[ruben@oc3463272252 AWS-EKS]$
```

Vista Web del Cluster creado:



paco-eks-cluster

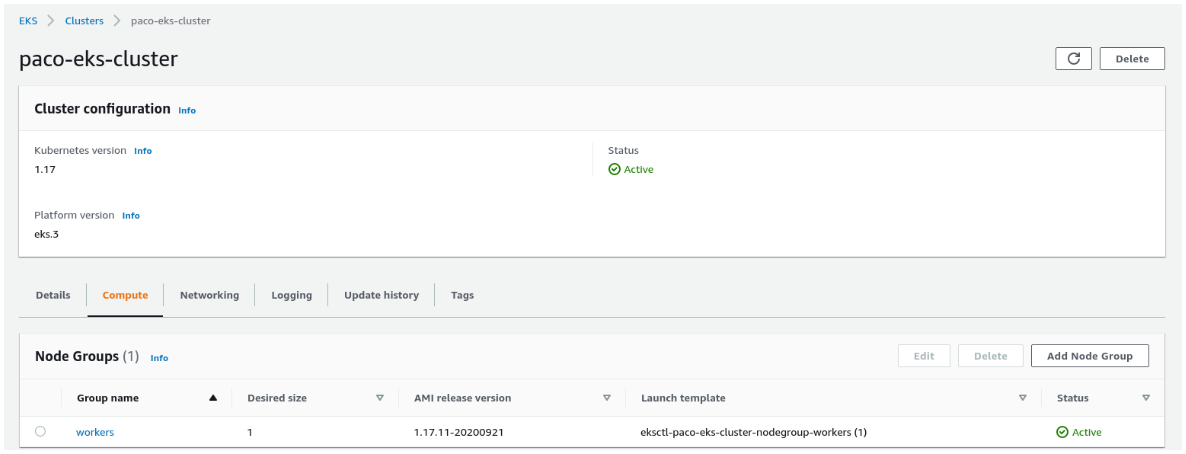
Cluster configuration Info

Kubernetes version Info	Status
1.17	✓ Active
Platform version Info	
eks.3	

Details

API server endpoint	OpenID Connect provider URL	Cluster ARN
https://82760354C708B396582069F13EAB0116.gr7.us-east-2.eks.amazonaws.com	https://oidc.eks.us-east-2.amazonaws.com/id/82760354C708B396582069F13EAB0116	arn:aws:eks:us-east-2:923176061796:cluster/paco-eks-cluster
Creation time	Certificate authority	Cluster IAM Role ARN
Sep 29th 2020 at 1:10 PM	LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUN5RENDQWJD Z0F3SUJBZ0lCQURBTkJna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VR WURWUVFERXdwcmRXSmwKY201bGRHVnpNQjRYRFRJd01Ea3lPV	arn:aws:iam::923176061796:role/eksctl-paco-eks-cluster-cluster-ServiceRole-C0SAL118TQ6

Mas detalles…



paco-eks-cluster

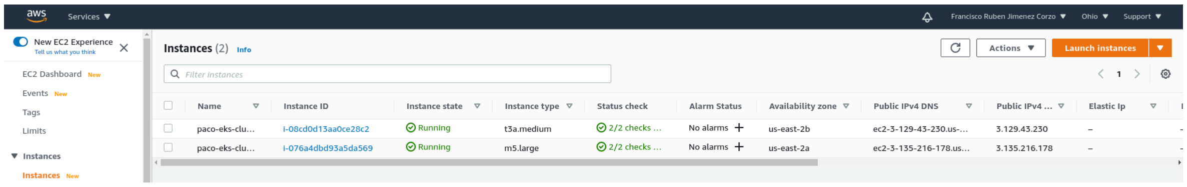
Cluster configuration Info

Kubernetes version Info	Status
1.17	✓ Active
Platform version Info	
eks.3	

Node Groups (1) Info

Group name	Desired size	AMI release version	Launch template	Status
workers	1	1.17.11-20200921	eksctl-paco-eks-cluster-nodegroup-workers (1)	✓ Active

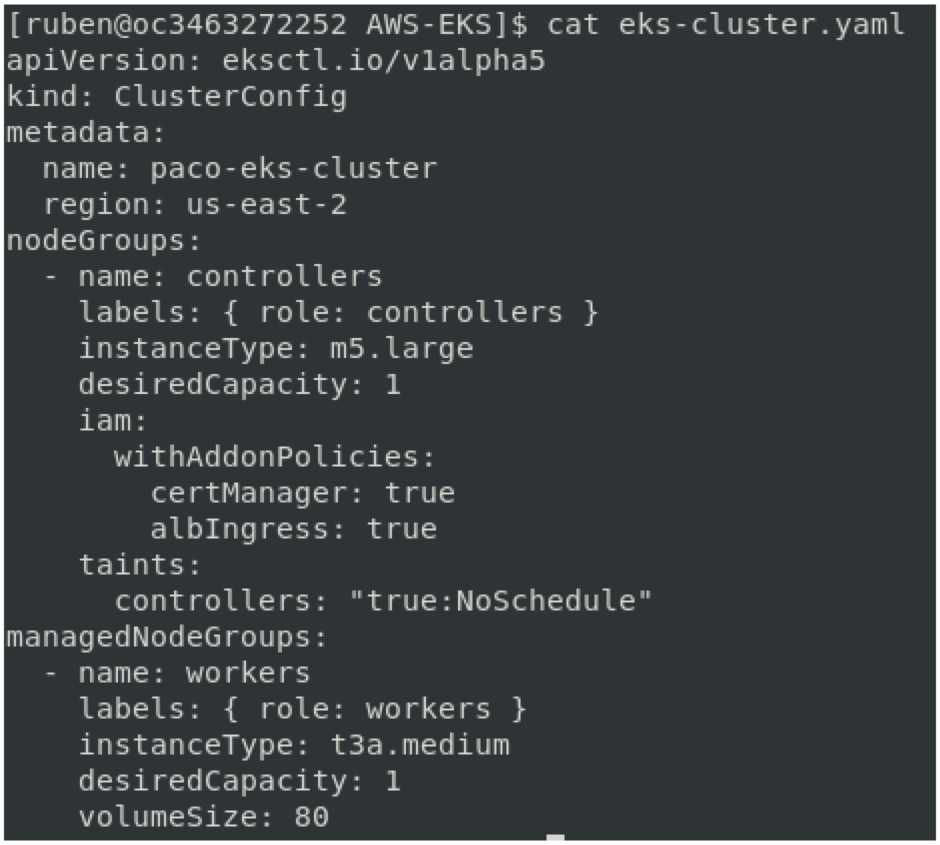
Vista de las EC2 Instancias:



Instances (2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm Status	Availability zone	Public IPv4 DNS	Public IPv4 …	Elastic Ip
paco-eks-clu…	i-08cd0d13aa0ce28c2	✓ Running	t3a.medium	✓ 2/2 checks …	No alarms +	us-east-2b	ec2-3-129-43-230.us-…	3.129.43.230	–
paco-eks-clu…	i-076a4dbd93a5da569	✓ Running	m5.large	✓ 2/2 checks …	No alarms +	us-east-2a	ec2-3-135-216-178.us…	3.135.216.178	–

Archivo YAML con los parametros de creacion del cluster:



```
[ruben@oc3463272252 AWS-EKS]$ cat eks-cluster.yaml
apiVersion: eksctl.io/v1alpha5
kind: ClusterConfig
metadata:
  name: paco-eks-cluster
  region: us-east-2
nodeGroups:
  - name: controllers
    labels: { role: controllers }
    instanceType: m5.large
    desiredCapacity: 1
    iam:
      withAddonPolicies:
        certManager: true
        albIngress: true
    taints:
      controllers: "true:NoSchedule"
managedNodeGroups:
  - name: workers
    labels: { role: workers }
    instanceType: t3a.medium
    desiredCapacity: 1
    volumeSize: 80
```

2. Instalar y Configurar el Nginx Ingress Controller
    1. kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.35.0/deploy/static/provider/aws/deploy.yaml

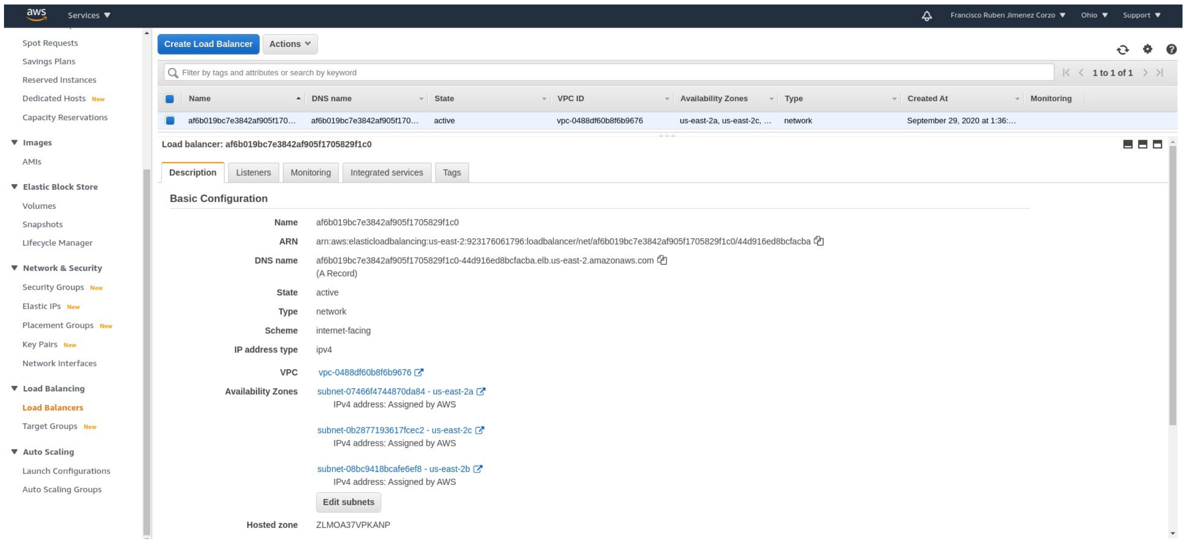
    2. Referece: https://kubernetes.github.io/ingress-nginx/deploy/#aws

In AWS: Network load balancer (NLB) to expose the NGINX Ingress controller behind a Service of Type=LoadBalancer.

3. Verificando:

```
$ kubectl get service/ingress-nginx-controller -n ingress-nginx

NAME                     TYPE           CLUSTER-IP      EXTERNAL-IP
PORT(S)                  AGE

ingress-nginx-controller   LoadBalancer   10.100.99.132   af6b019bc7e3842af905f1705829f1c0-
44d916ed8bcfacba.elb.us-east-2.amazonaws.com   80:30565/TCP,443:30510/TCP   3h46m

$
```

Detalle del Load Balancer creado:



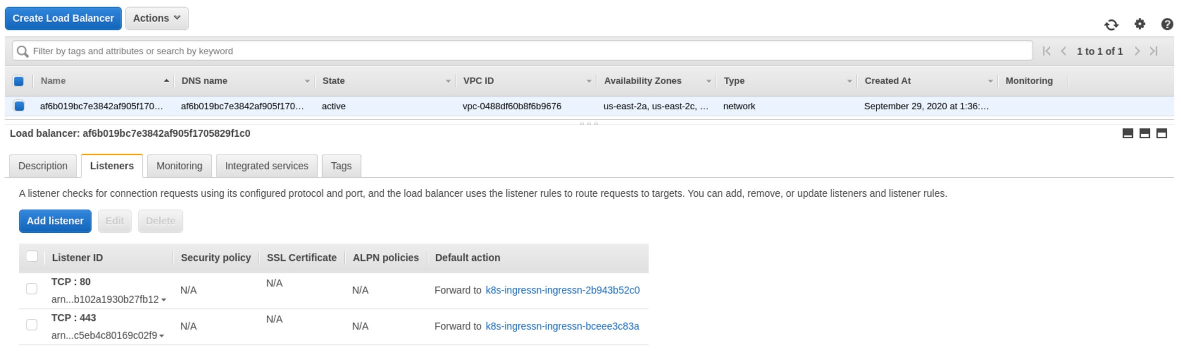
Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring
af6b019bc7e3842af905f170…	af6b019bc7e3842af905f170…	active	vpc-0488df60b8f6b9676	us-east-2a, us-east-2c, …	network	September 29, 2020 at 1:36:…	

Load balancer: af6b019bc7e3842af905f1705829f1c0

Basic Configuration

Name	af6b019bc7e3842af905f1705829f1c0
ARN	arn:aws:elasticloadbalancing:us-east-2:923176061796:loadbalancer/net/af6b019bc7e3842af905f1705829f1c0/44d916ed8bcfacba
DNS name	af6b019bc7e3842af905f1705829f1c0-44d916ed8bcfacba.elb.us-east-2.amazonaws.com (A Record)
State	active
Type	network
Scheme	internet-facing
IP address type	ipv4
VPC	vpc-0488df60b8f6b9676
Availability Zones	subnet-07466f4744870da84 - us-east-2a IPv4 address: Assigned by AWS subnet-0b2877193617fcec2 - us-east-2c IPv4 address: Assigned by AWS subnet-08bc9418bcafe6ef8 - us-east-2b IPv4 address: Assigned by AWS
Hosted zone	ZLMOA37VPKANP

Mas detalles…



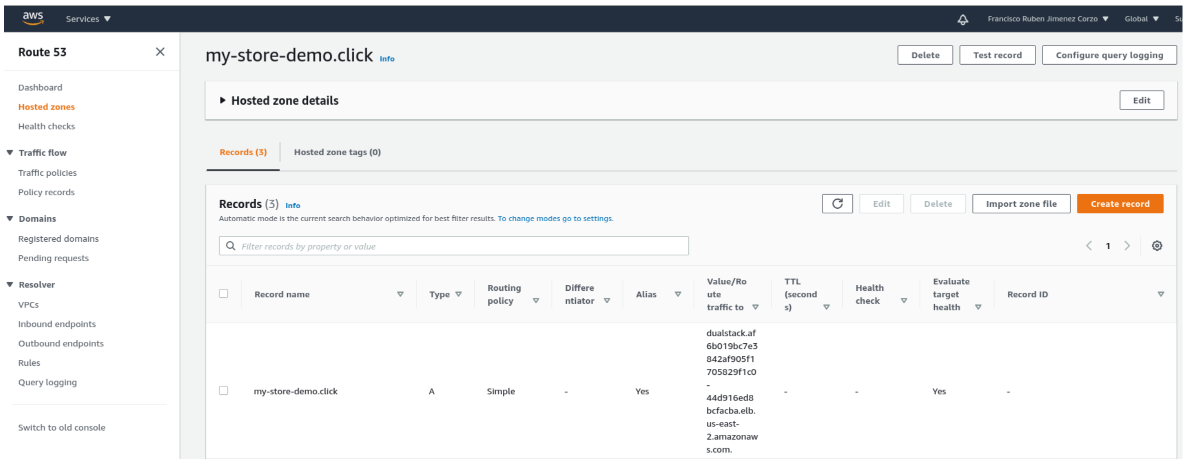
Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring
af6b019bc7e3842af905f170…	af6b019bc7e3842af905f170…	active	vpc-0488df60b8f6b9676	us-east-2a, us-east-2c, …	network	September 29, 2020 at 1:36:…	

Load balancer: af6b019bc7e3842af905f1705829f1c0

A listener checks for connection requests using its configured protocol and port, and the load balancer uses the listener rules to route requests to targets. You can add, remove, or update listeners and listener rules.

Listener ID	Security policy	SSL Certificate	ALPN policies	Default action
TCP : 80 arn...b102a1930b27fb12	N/A	N/A	N/A	Forward to k8s-ingressn-ingressn-2b943b52c0
TCP : 443 arn...c5eb4c80169c02f9	N/A	N/A	N/A	Forward to k8s-ingressn-ingressn-bceee3c83a

Nota: Desde Route53 hay que crear un Record de tipo A, para redireccionar el sitio: http://my-store-demo.click a el NLB (Network Load Balancer) creado por el servicio creado: af6b019bc7e3842af905f1705829f1c0-44d916ed8bcfacba.elb.us-east-2.amazonaws.com



my-store-demo.click

Records (3)

Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

Record name	Type	Routing policy	Differentiator	Alias	Value/Route traffic to	TTL (seconds)	Health check	Evaluate target health	Record ID
my-store-demo.click	A	Simple	-	Yes	dualstack.af6b019bc7e3842af905f1705829f1c0-44d916ed8bcfacba.elb.us-east-2.amazonaws.com.	-	-	Yes	-

3. Instalar el servicio Cert-Manager:
   (1) kubectl apply --validate=false -f https://github.com/jetstack/cert-manager/releases/download/v1.0.2/cert-manager.yaml

   (2) Reference: https://cert-manager.io/docs/installation/kubernetes/

   (3) Verificando: $ kubectl get pods --namespace cert-manager

```
NAME                                       READY   STATUS    RESTARTS   AGE
cert-manager-74fdd7bc8f-k72rz              1/1     Running   0          3h32m
cert-manager-cainjector-8665f9998-vc798    1/1     Running   0          3h32m
cert-manager-webhook-677b8cbf67-6pt96      1/1     Running   0          3h32m
$
```

4. Creacion de los Backends e Ingress sin manejo de certificados:
   - kubectl create -f all-store.yaml

   - kubectl create -f ingress-demo-unsecure.yaml

```
$ kubectl describe ingress ingress-demo
Name:             ingress-demo
Namespace:        default
Address:          af6b019bc7e3842af905f1705829f1c0-44d916ed8bcfacba.elb.us-east-2.amazonaws.com
Default backend:  page404:80 (192.168.26.30:80)
Rules:
  Host                Path  Backends
  ----                ----  --------
  my-store-demo.click
                      /store   store:80 (192.168.2.238:80)
                      /cart    cart:80 (192.168.24.67:80)
                      /home    home:80 (192.168.28.129:80)
                      /        home:80 (192.168.28.129:80)
                      /(.+)    page404:80 (192.168.26.30:80)
Annotations:          cert-manager.io/cluster-issuer: letsencrypt-prod
                      kubernetes.io/ingress.class: nginx
                      nginx.ingress.kubernetes.io/rewrite-target: /
```

```
$ kubectl get all -o wide

NAME        READY   STATUS    RESTARTS   AGE     IP             NODE
NOMINATED NODE   READINESS GATES

pod/cart      1/1     Running   0          3h53m   192.168.24.67    ip-192-168-28-169.us-east-2.compute.internal
<none>          <none>

pod/home      1/1     Running   0          167m    192.168.28.129   ip-192-168-28-169.us-east-2.compute.internal
<none>          <none>

pod/page404   1/1     Running   0          3h53m   192.168.26.30    ip-192-168-28-169.us-east-
2.compute.internal   <none>          <none>

pod/store     1/1     Running   0          3h53m   192.168.2.238    ip-192-168-28-169.us-east-2.compute.internal
<none>          <none>


NAME                 TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)   AGE     SELECTOR

service/cart         ClusterIP   10.100.119.14    <none>        80/TCP    3h53m   run=cart

service/home         ClusterIP   10.100.140.133   <none>        80/TCP    3h53m   run=home

service/kubernetes   ClusterIP   10.100.0.1       <none>        443/TCP   4h13m   <none>

service/page404      ClusterIP   10.100.85.112    <none>        80/TCP    3h53m   run=page404

service/store        ClusterIP   10.100.138.117   <none>        80/TCP    3h53m   run=store

$
```

5. Creat el certificate issuer, (staging)
   `kubectl create -f staging_issuer.yaml`

6. Aplicar la configuracion de TLS al ingress, y verificacion
   (1) `kubectl apply -f ingress-demo.yaml`

   (2) `kubectl describe certificate`

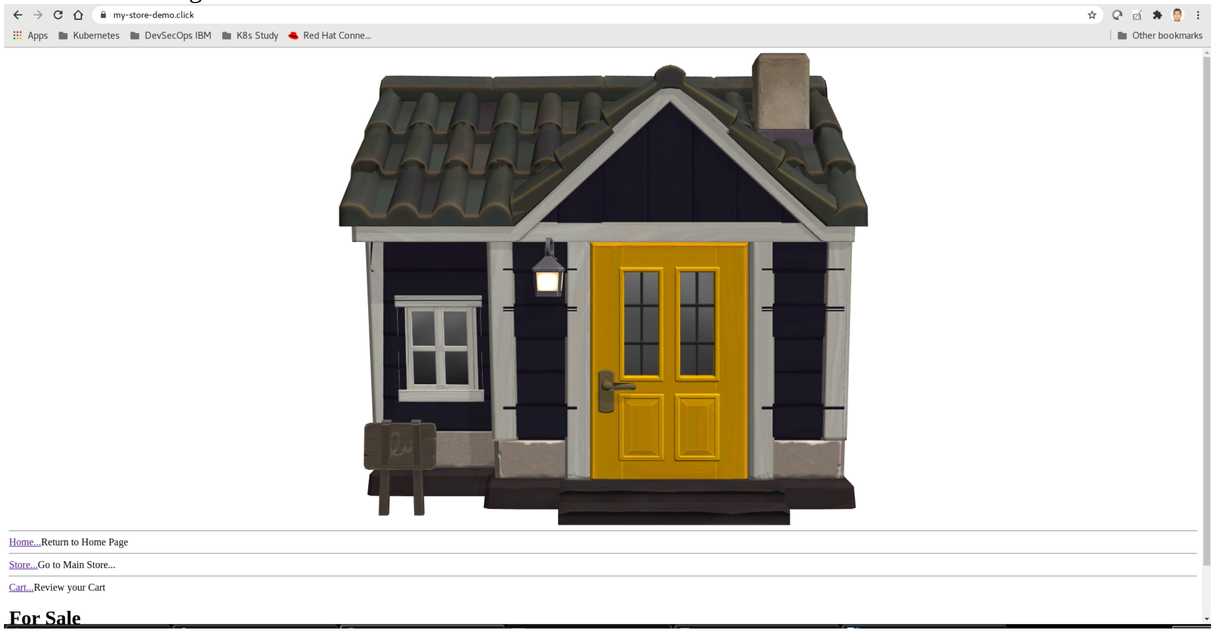
```
$ kubectl describe certificate

Name:         store-demo-tls
Namespace:    default
Labels:       <none>
Annotations:  <none>
API Version:  cert-manager.io/v1
Kind:         Certificate
Metadata:
  Creation Timestamp:  2020-09-29T18:59:28Z
  Generation:          1
  Owner References:
    API Version:           extensions/v1beta1
    Block Owner Deletion:  true
    Controller:           true
    Kind:                 Ingress
    Name:                 ingress-demo
    UID:                  13681da2-7657-4072-abc0-5852c72a4eea
  Resource Version:        8962
  Self Link:               /apis/cert-manager.io/v1/namespaces/default/certificates/store-demo-tls
  UID:                     35e441a5-5f76-4891-9fad-66d0af38a694
Spec:
  Dns Names:
    my-store-demo.click
  Issuer Ref:
    Group:      cert-manager.io
    Kind:       ClusterIssuer
    Name:       letsencrypt-staging
  Secret Name:  store-demo-tls
Status:
  Conditions:
    Last Transition Time:  2020-09-29T18:59:52Z
    Message:               Certificate is up to date and has not expired
    Reason:                Ready
    Status:                True
    Type:                  Ready
  Not After:               2020-12-28T17:59:52Z
  Not Before:              2020-09-29T17:59:52Z
  Renewal Time:            2020-11-28T17:59:52Z
  Revision:                1
Events:
  Type    Reason     Age   From          Message
  ----    ------     ----  ----          -------
  Normal  Issuing    104s  cert-manager  Issuing certificate as Secret does not exist
  Normal  Generated  104s  cert-manager  Stored new private key in temporary Secret resource "store-demo-tls-wpzjx"
  Normal  Requested  104s  cert-manager  Created new CertificateRequest resource "store-demo-tls-frdrx"
  Normal  Issuing    81s   cert-manager  The certificate has been successfully issued
$
```

- Verificando desde linea de comandos:

```
$ curl http://my-store-demo.click
<html>
<head><title>308 Permanent Redirect</title></head>
<body>
<center><h1>308 Permanent Redirect</h1></center>
```

```
<hr><center>nginx/1.19.2</center>
</body>
</html>
$
```

- Verificando desde Navegador Web:
  - Pagina de Home de la Store de Demo:



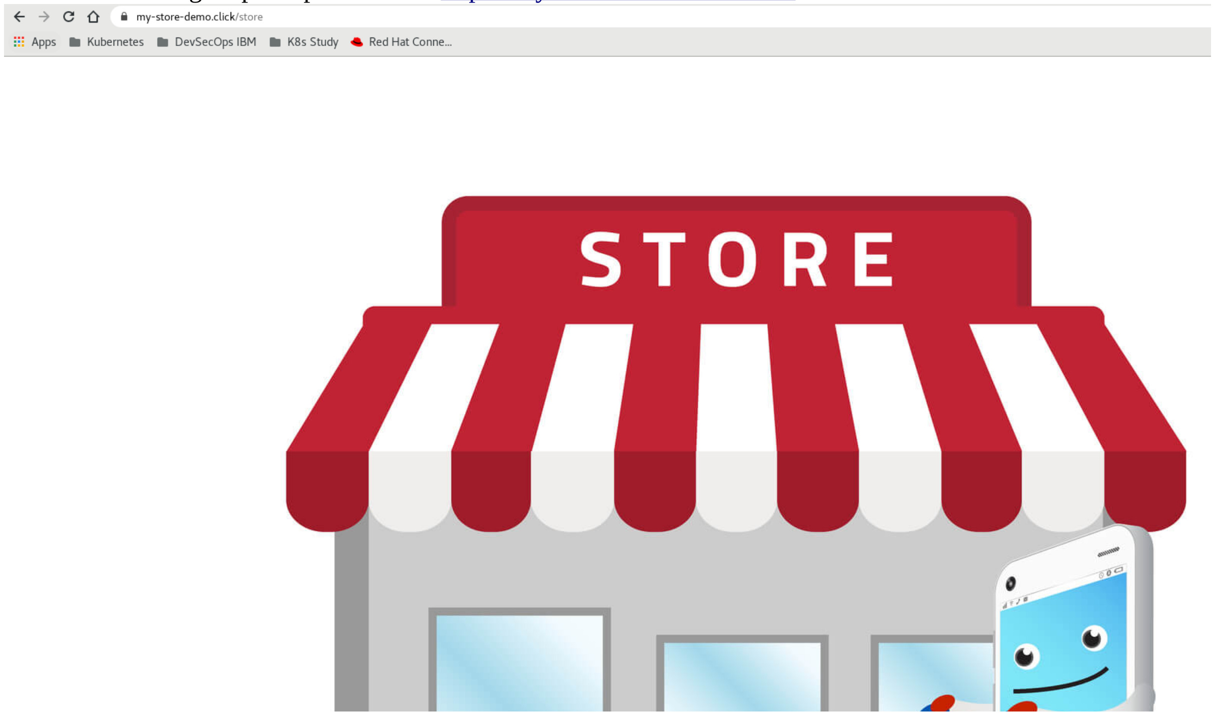
Home...Return to Home Page

Store...Go to Main Store...

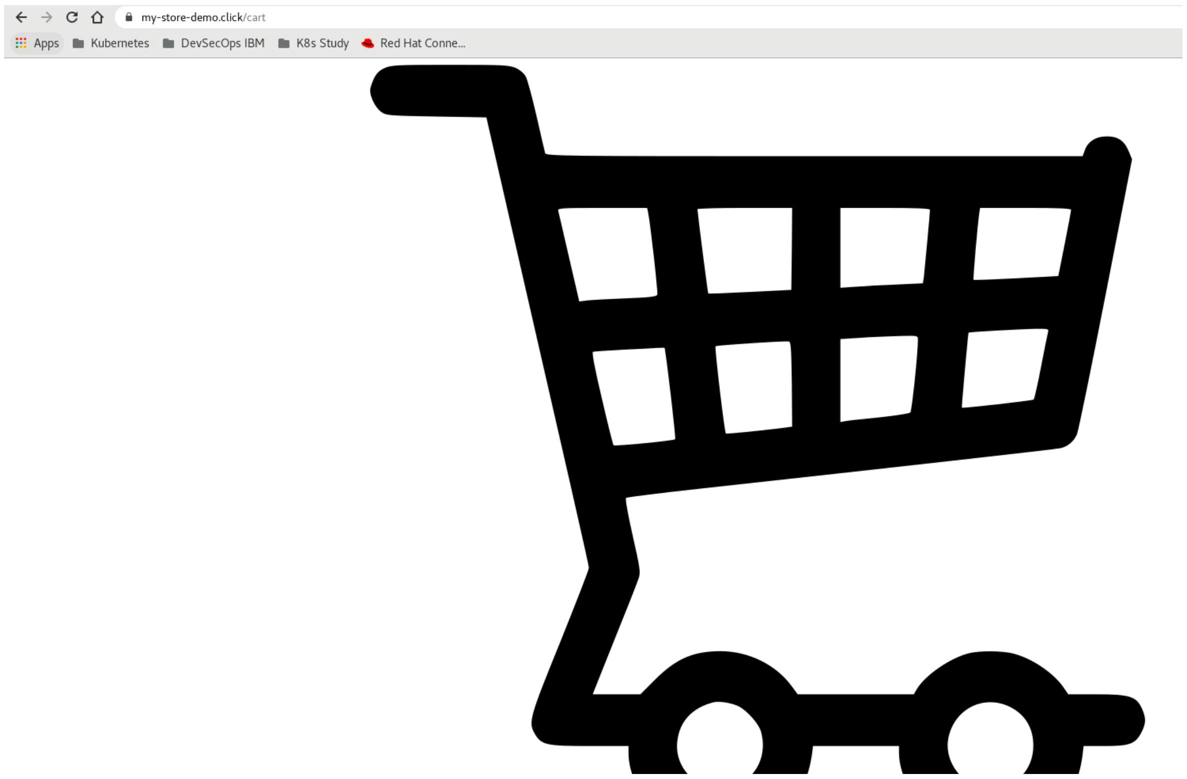
Cart...Review your Cart

For Sale

  - Pagina principal de store: https://my-store-demo.click/store

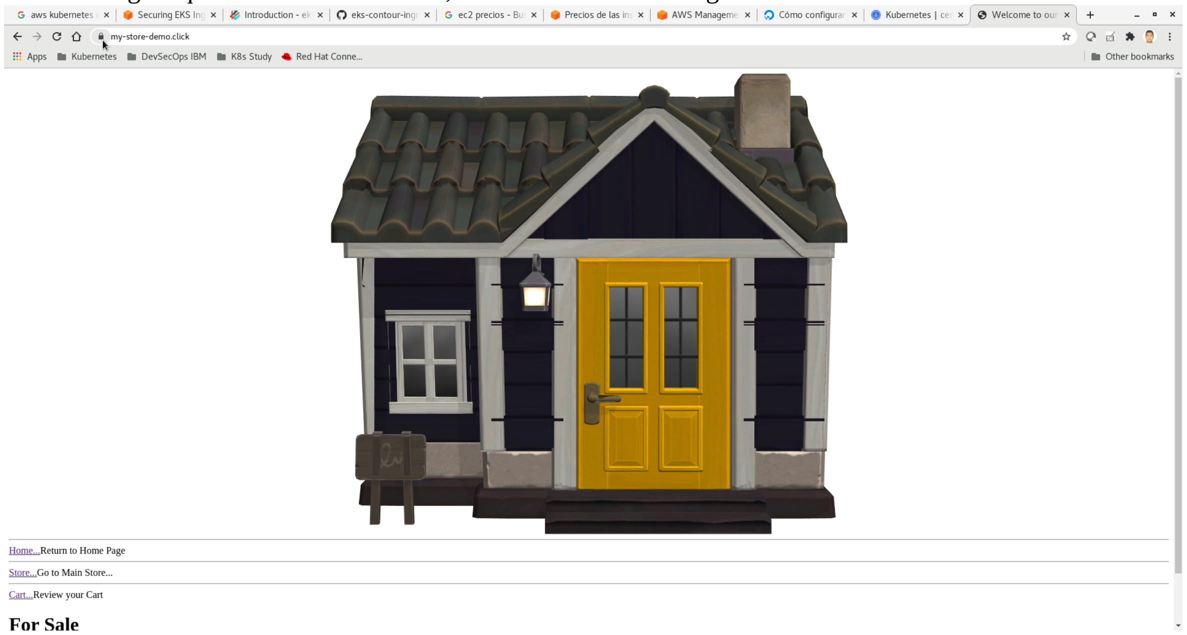


  - Pagina del Carrito de Compras: https://my-store-demo.click/cart


my-store-demo.click/cart

Apps Kubernetes DevSecOps IBM K8s Study Red Hat Conne...

- Pagina que muestra el candadito, indicador de un sitio seguro con certificado autentico:



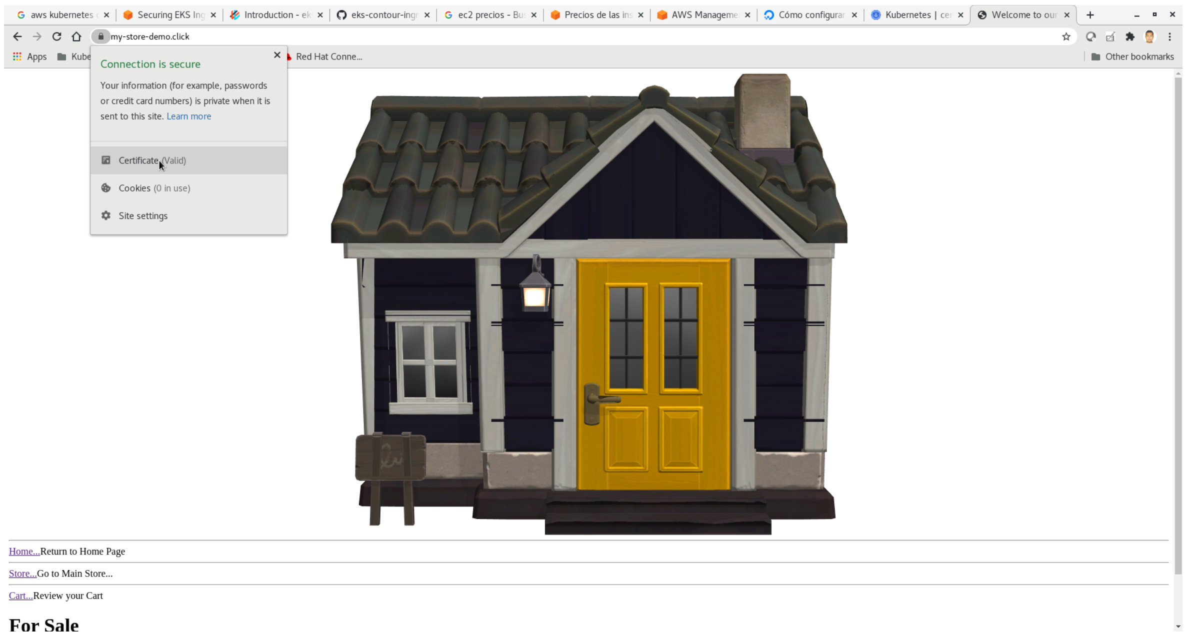
Home...Return to Home Page

Store...Go to Main Store...

Cart...Review your Cart

For Sale

- Detalle del Certificado valido del sitio seguro: https://my-store-demo.click/



Connection is secure

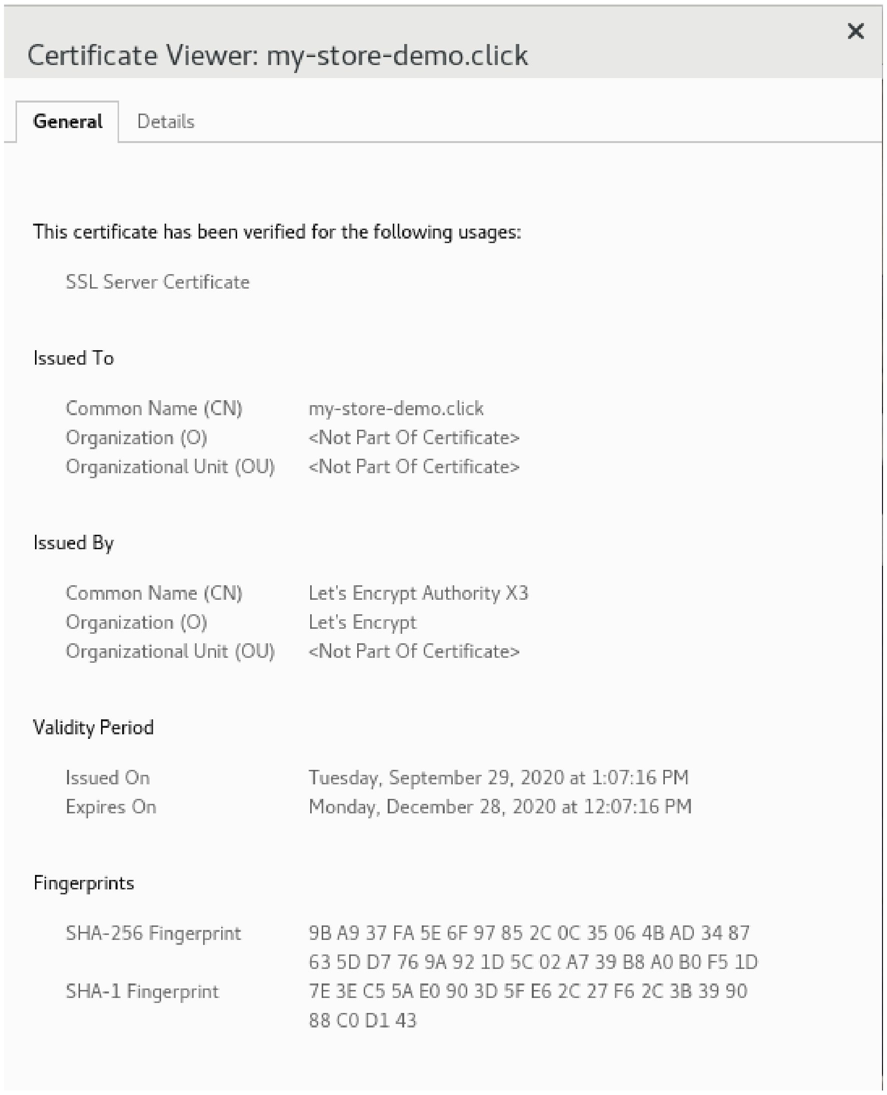
Your information (for example, passwords or credit card numbers) is private when it is sent to this site. Learn more

Home...Return to Home Page

Store...Go to Main Store...

Cart...Review your Cart

For Sale



Certificate Viewer: my-store-demo.click

This certificate has been verified for the following usages:

SSL Server Certificate

Issued To

Common Name (CN)	my-store-demo.click
Organization (O)	<Not Part Of Certificate>
Organizational Unit (OU)	<Not Part Of Certificate>

Issued By

Common Name (CN)	Let's Encrypt Authority X3
Organization (O)	Let's Encrypt
Organizational Unit (OU)	<Not Part Of Certificate>

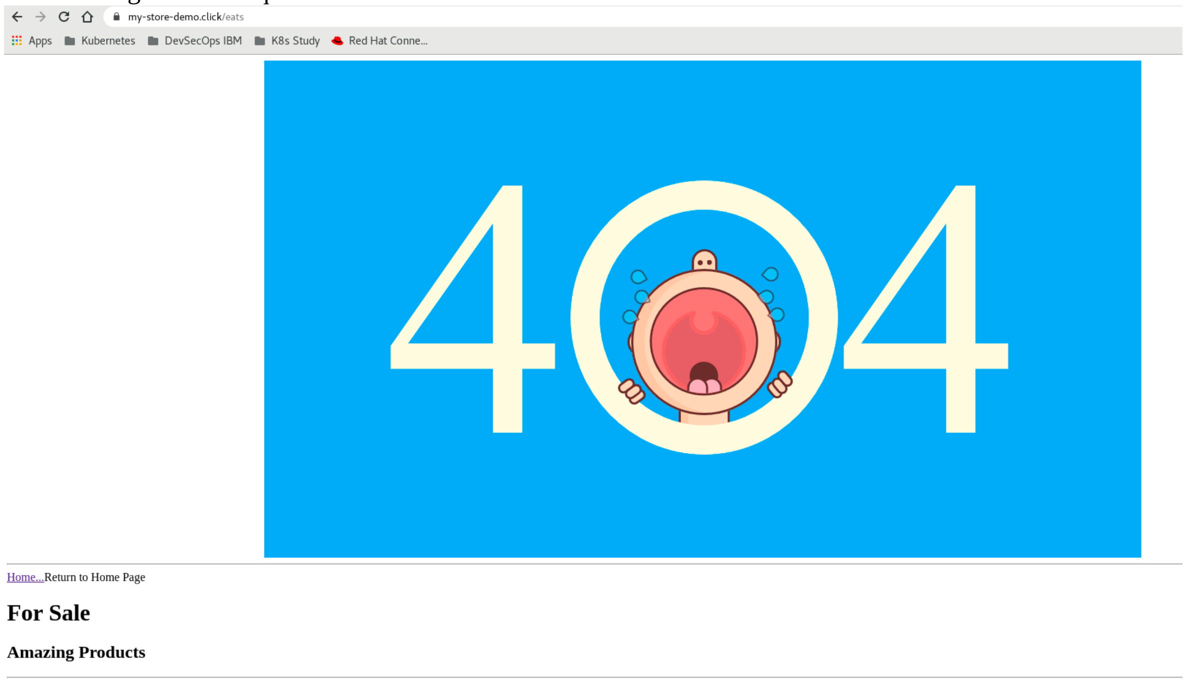
Validity Period

Issued On	Tuesday, September 29, 2020 at 1:07:16 PM
Expires On	Monday, December 28, 2020 at 12:07:16 PM

Fingerprints

SHA-256 Fingerprint	9B A9 37 FA 5E 6F 97 85 2C 0C 35 06 4B AD 34 87 63 5D D7 76 9A 92 1D 5C 02 A7 39 B8 A0 B0 F5 1D
SHA-1 Fingerprint	7E 3E C5 5A E0 90 3D 5F E6 2C 27 F6 2C 3B 39 90 88 C0 D1 43

- Pagina de cualquier otro sitio:



Home...Return to Home Page

For Sale

Amazing Products

7. Listado de temas a visitar:

- aws-cli
- Kubernetes
- EKS
- eksctl
- YAML Files
- EC2 Instances
- Nginx Ingress Controller
- Container
- InitContainer
- Pod
- Services
- Deployments
- Ingress
- Cert-Manager
- TLS Certificados
- Let's Encrypt

8. Otros posibles escenarios similares o mejoras:

(1) Securing EKS Ingress With Contour And Let’s Encrypt The GitOps Way
(2) Emplear Spot Instances para este mismo demo, para bajar el costo del EKS cluster
(3) Emplear un Cluster de Kubernetes corriendo en Baremetal on WAS

9. Mas Detalles, Contacto:

Francisco Ruben Jimenez Corzo
rubenj@mx1.ibm.com