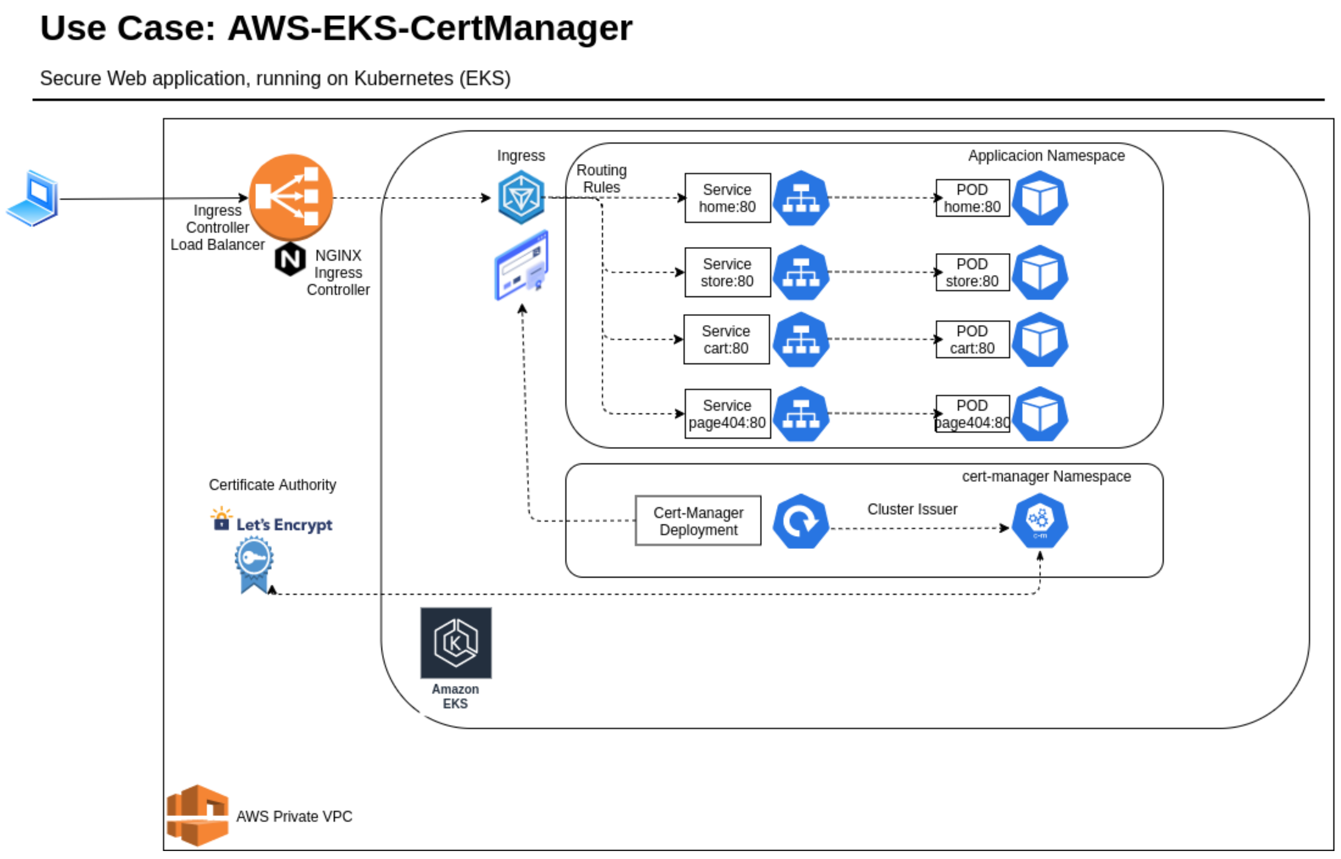
Github repository: https://github.com/jimenezcorzo/AWS-EKS-CertManager

Use Case: Secure Web Application running on AWS EKS, https://my-store-demo.click

Short description:

Create a Simple Web Application hosted on Managed Kubernetes Services EKS on AWS Cloud Provider, configure the whole architecture to support https (TLS Certificate) valid trough a Certificate Authority (Let's Encrypt), on domain: https://my-store-demo.click

Architecture diagram:



General Procedure:

1. Create EKS Cluster (Managed Kubernetes)
2. Verify cluster access, trought kubectl command
3. Install Nginx Controller for AWS Kubernetes
4. Verify controller with command:
   - kubectl get service/ingress-nginx-controller -n ingress-nginx

5. Add Alias (A) Record on Route53 to point to LoadBalancer
6. Install Cert-Manager for AWS Kubernetes
7. Verify the installation with command:
   - kubectl get pods --namespace cert-manager

8. Create the Web Application (Service Backends)
9. Verify with command:
   - kubectl get pod,svc -o wide
10. Create the Ingress (No TLS)
11. Verify with command:
    - kubectl describe ingresses ingress-demo

12. Create Certificate Issuer (staging)
13. Verify with command:
    - kubectl describe certificate

14. Modify the Ingress to add TLS certificate security
15. Verify with command:
    - kubectl describe ingresses ingress-demo

16. Replace Certificate Issues, from staging to prod
17. Verify with command:
    - kubectl describe certificate

18. Navigate the web site: https://my-store-demo.click/

Concepts visited on this procedure:

- 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

Contact if you need further details about this use case:
Francisco Ruben Jimenez Corzo
rubenj@mx1.ibm.com