Kubernetes Ingress

Monday, October 20, 2025

5:24 PM

**ALB vs. Nginx Ingress**

**===================**

**There are different Ingress Classes that can be configured on Kubernetes.**

1. **ALB**
   1. **Ex:**

1. **Nginx**
   1. **Nginx is being used when we want to configure ONE Load Balancer for different services in different namespaces.**
   2. **In this case, it is being provisioned by the Kubernetes nginx-controller and it will handle all the requests for all the services under the same ALB Load Balancer.**
   3. **Notice that, all of these domains need to be reachable on HTTPS not HTTP.**
   4. **Currently, all of these domains share the same ALB load balancer, because it was provisioned by NGINX Ingress controller in the ingress-nginx namespace.** 
      1. **Services Example:**
         * [**https://ui-uads-dev.aw.lz.us-cert.gov/**](https://ui-uads-dev.aw.lz.us-cert.gov/)

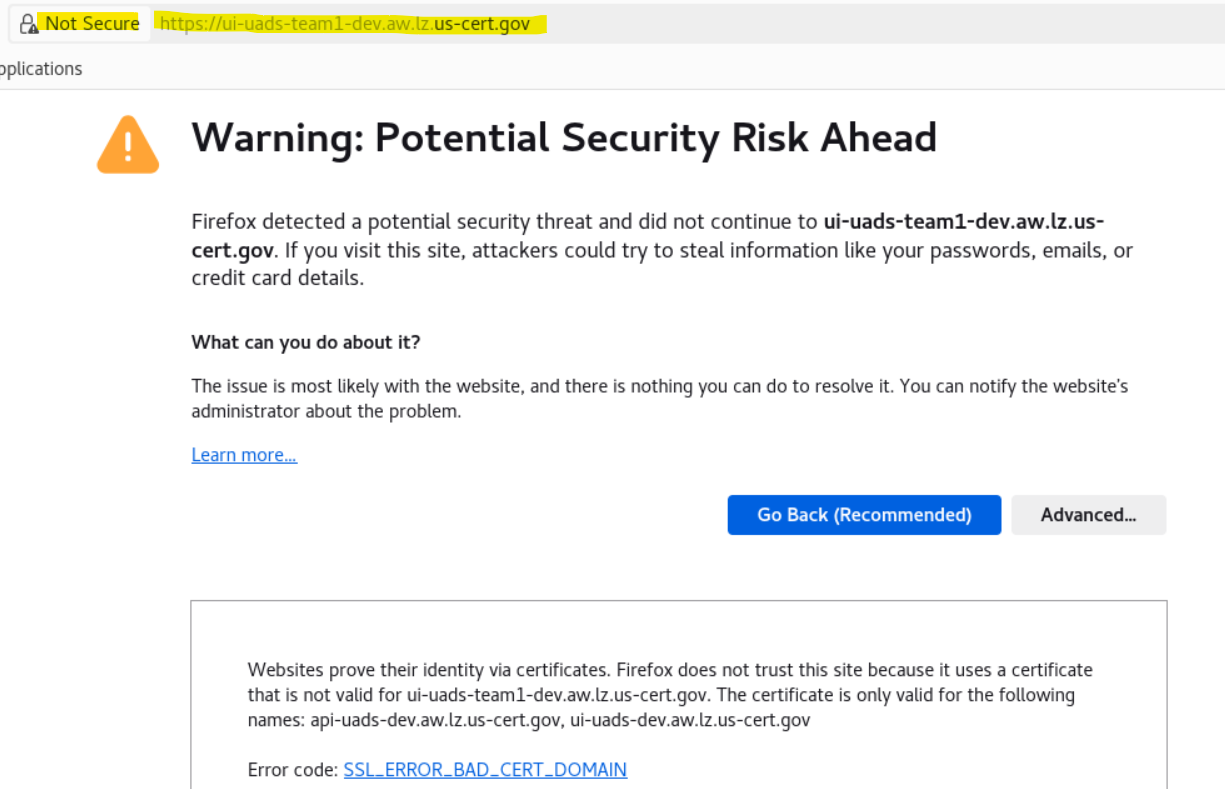
* [**https://ui-uads-test.aw.lz.us-cert.gov/**](https://ui-uads-test.aw.lz.us-cert.gov/)

* [**https://ui-uads-team1-dev.aw.lz.us-cert.gov/**](https://ui-uads-team1-dev.aw.lz.us-cert.gov/)

* [**https://ui-hedwig-dev.aw.lz.us-cert.gov/**](https://ui-hedwig-dev.aw.lz.us-cert.gov/)

**Error:**

**This is an example for trying to access** [**https://ui-uads-team1-dev.aw.lz.us-cert.gov/**](https://ui-uads-team1-dev.aw.lz.us-cert.gov/)**, without having the proper certificate installed on the NGINX ALB Load Balancer**.

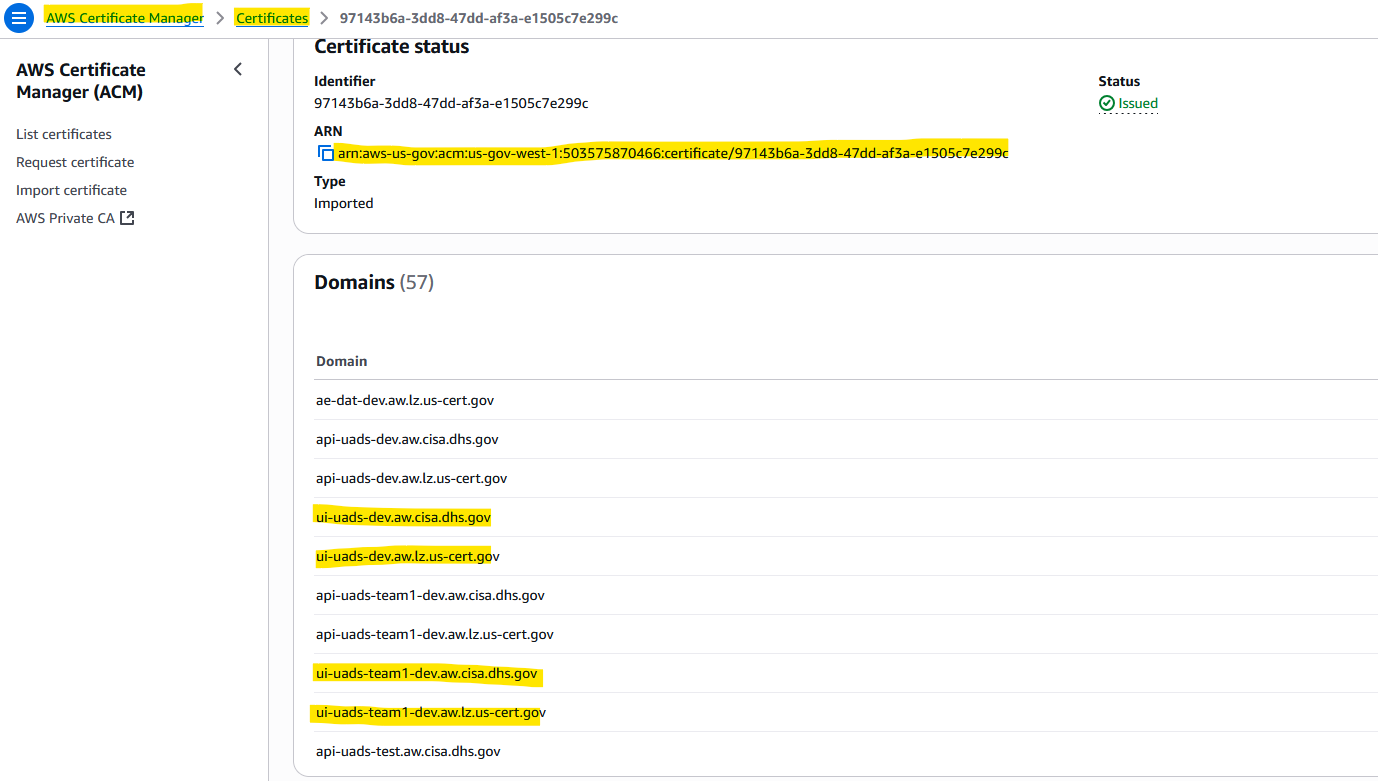


**Solution:**

The **load balancer certificate** need to include **all the domains** to be accessed through this **NGINX ALB load balancer**.

Then, it need to be **manually configured** on the **ingress-nginx controller** in the **ingress-nginx namespace** in the **aw-nonprod-eks-cluster**

**AWS Certificate Manager: (ACM)**



**Ingress-Nginx Controller:**

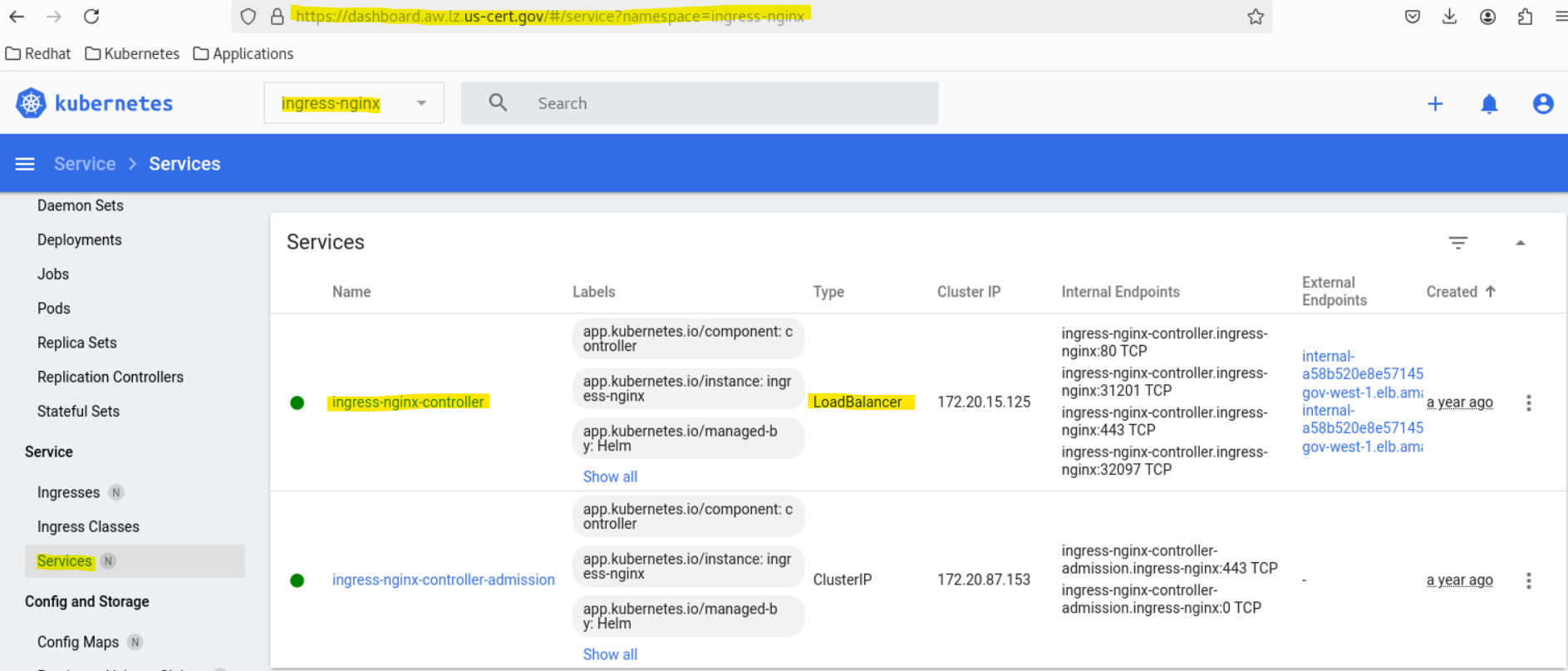
I had to **change** the **certificate** from the **old value** which wasn't found in **ACM**, to the show value below which **include all the domains** for **UADS**.

**Cluster: aw-nonprod-eks-cluster**

**Namespace: ingress-nginx**

**Services: ingress-nginx-controller**

**Ingress-nginx-controller**



**This is showing the load balancer that is being provisioned by the ingress-nginx controller.**

**ALB Load balancer: internal-a58b520e8e57145339083bbbec672da1-941152249.us-gov-west-1.elb.amazonaws.com**

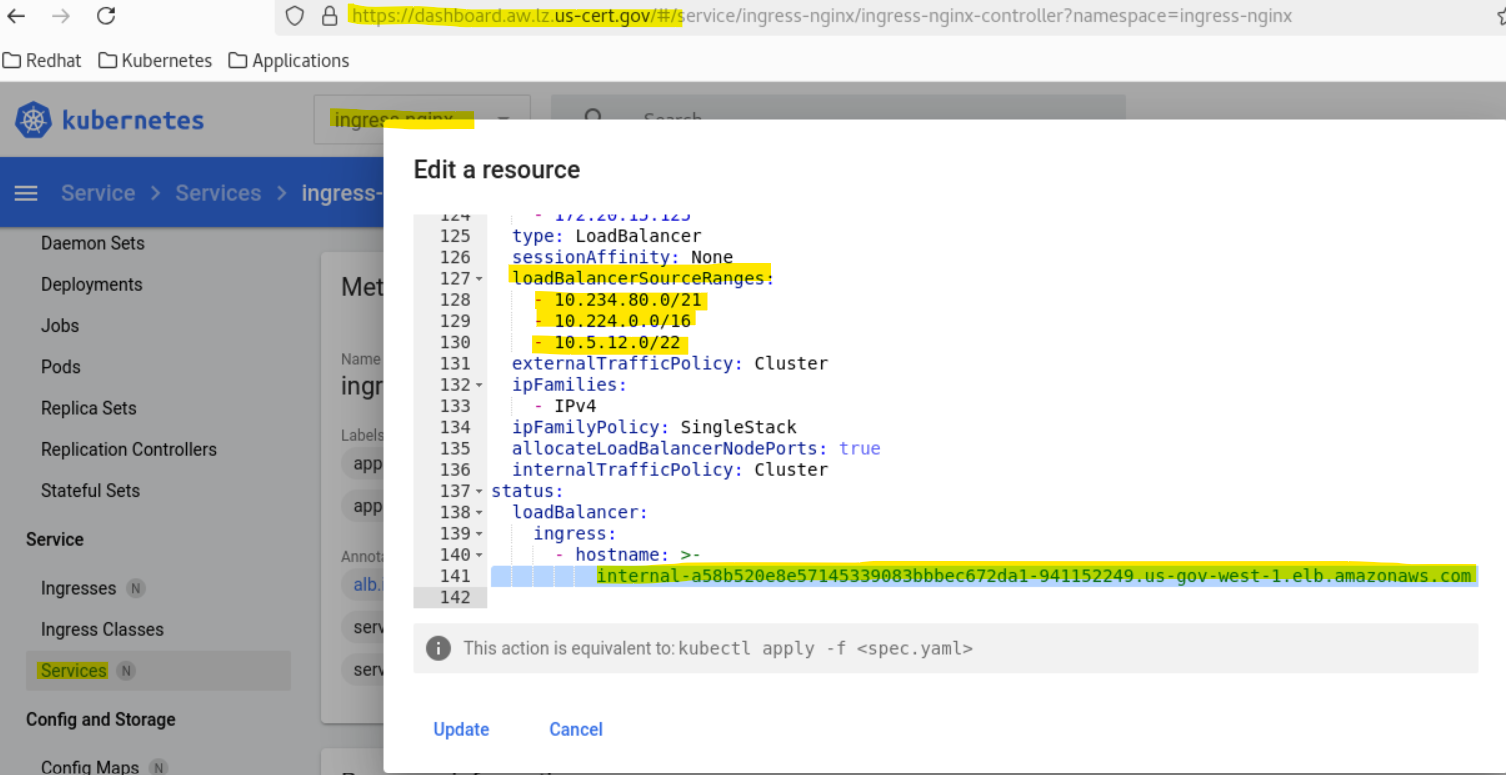
**Notice that, you can change the security group for the ALB load balancer and add any new CIDR ranges that are required to be added for this ALB load balancer from the loadBalancerSourceRanges highlighted below.**

**loadBalancerSourceRanges:**

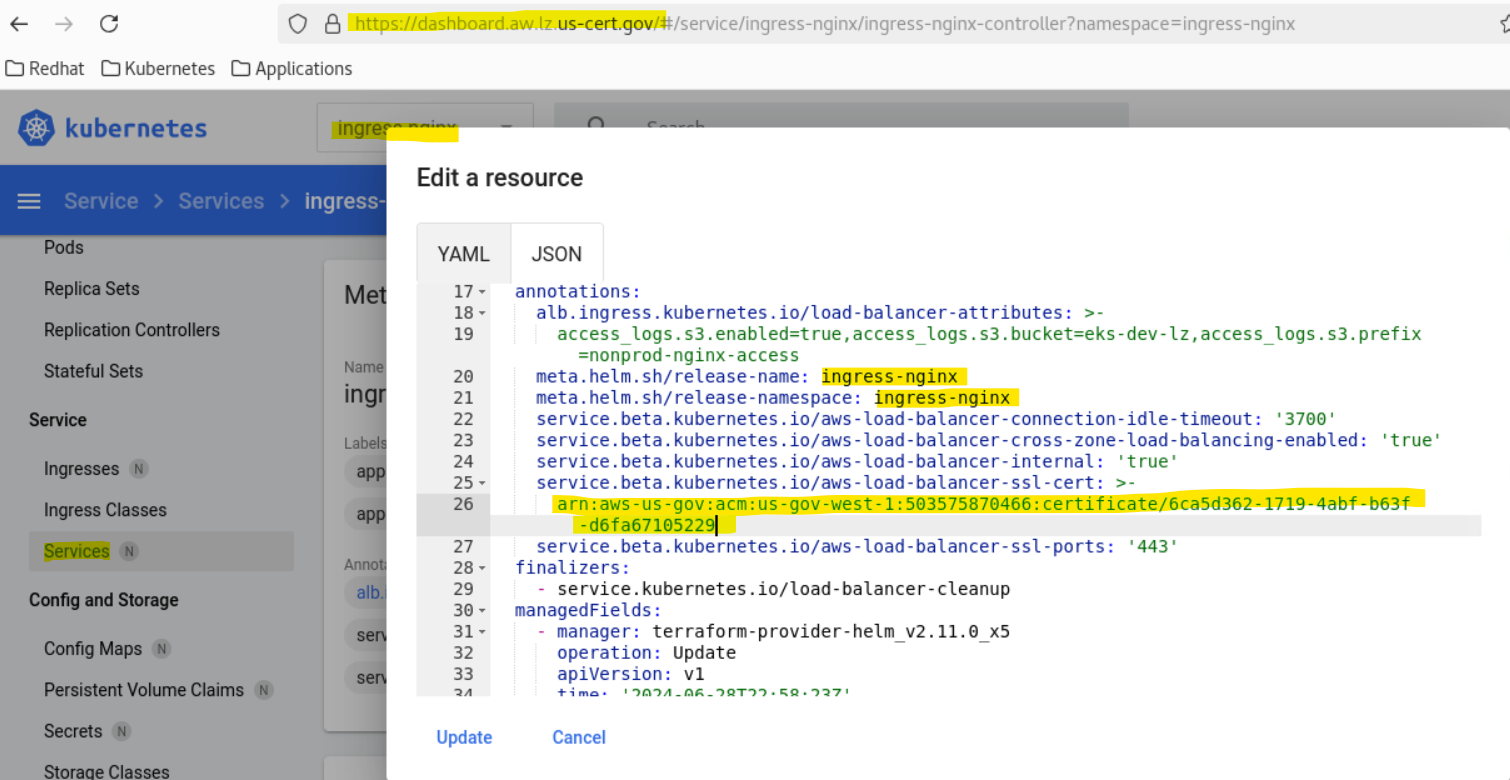
**- 10.234.80.0/21 ---> This is to make them accessible through our Redhat Workspaces. (**This is the **Subnet** of the **VPC** that hosts all our **AWS Workspaces)**

**- 10.224.0.0/16**

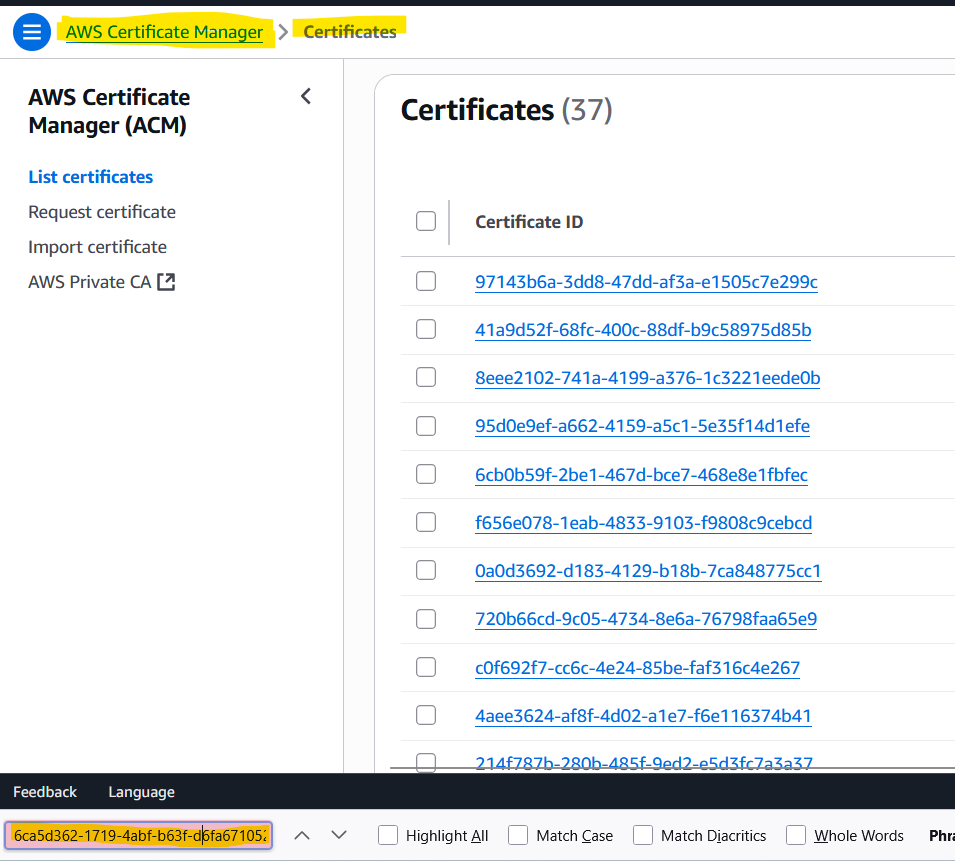
**- 10.5.12.0/22**



**OLD Incorrect Value for Certificate:**



**Notice that it doesn't exist in ACM, and that's why the domains aren't reachable on HTTPS.**



**All Unreachable Domains:**

**Notice that, all of these domains need to be reachable on HTTPS not HTTP.**

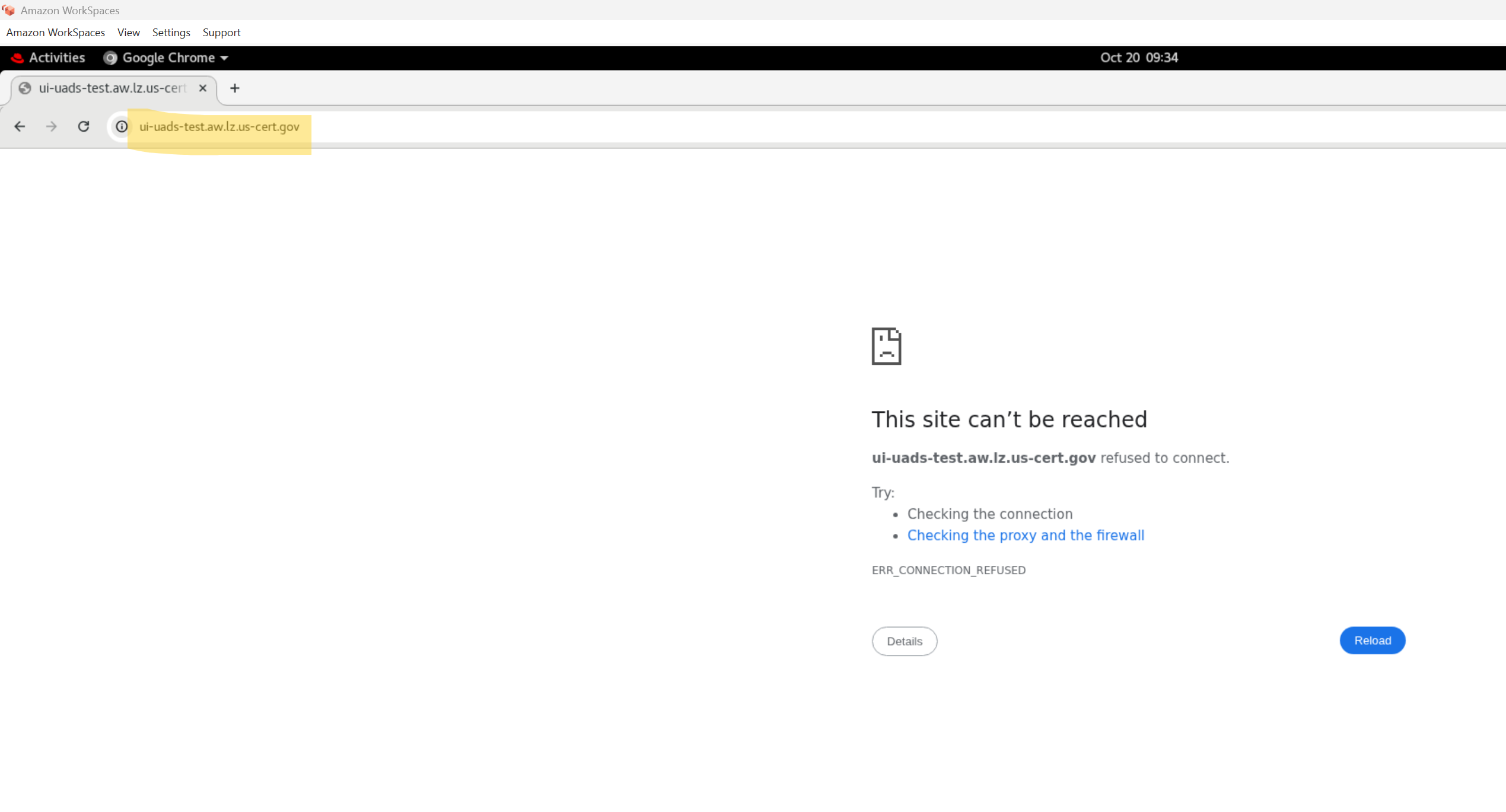
* [**https://ui-uads-dev.aw.lz.us-cert.gov/**](https://ui-uads-dev.aw.lz.us-cert.gov/)

* [**https://ui-uads-test.aw.lz.us-cert.gov/**](https://ui-uads-test.aw.lz.us-cert.gov/)

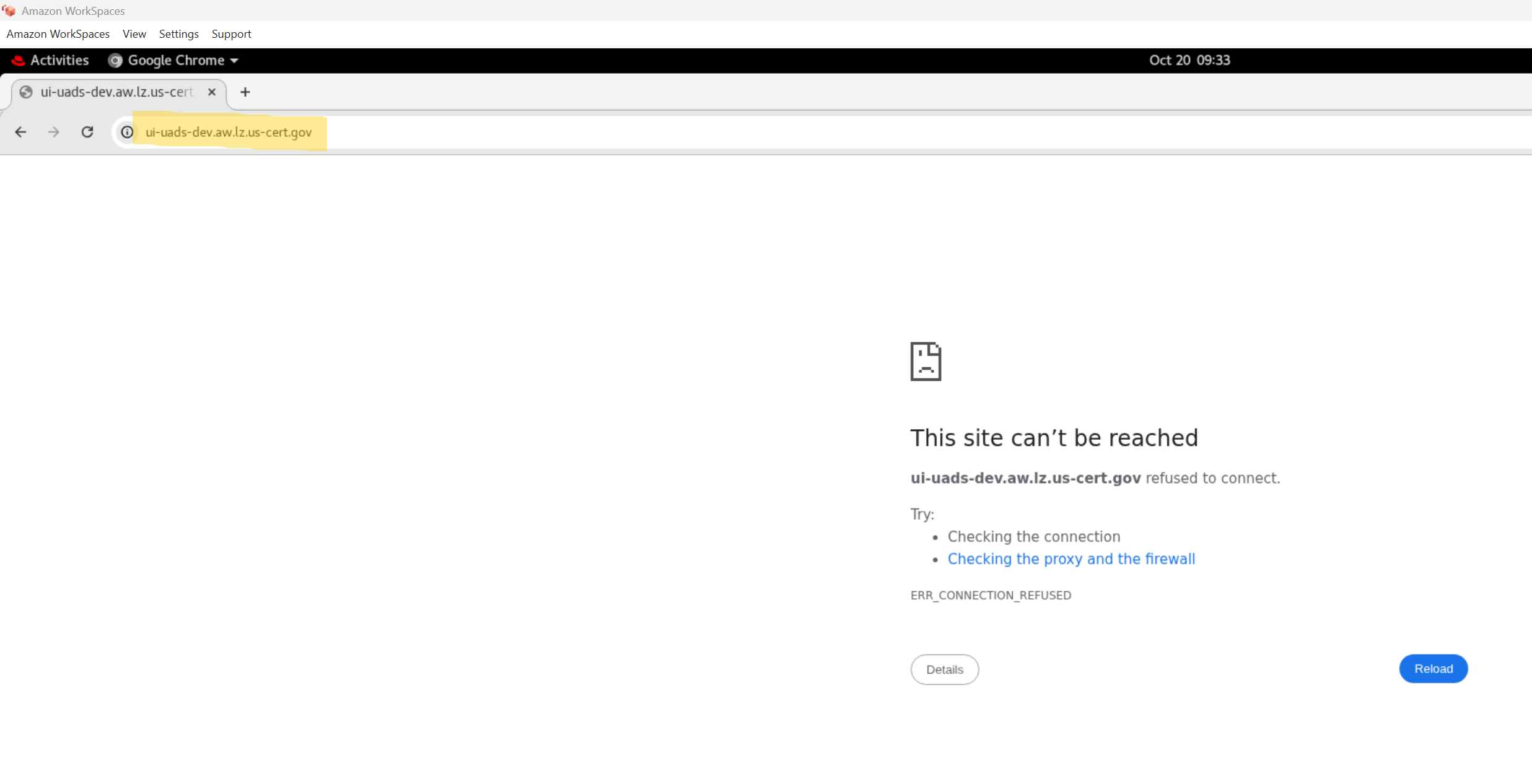
* [**https://ui-uads-team1-dev.aw.lz.us-cert.gov/**](https://ui-uads-team1-dev.aw.lz.us-cert.gov/)

* [**https://ui-hedwig-dev.aw.lz.us-cert.gov/**](https://ui-hedwig-dev.aw.lz.us-cert.gov/)

[**https://ui-uads-test.aw.lz.us-cert.gov/**](https://ui-uads-test.aw.lz.us-cert.gov/)

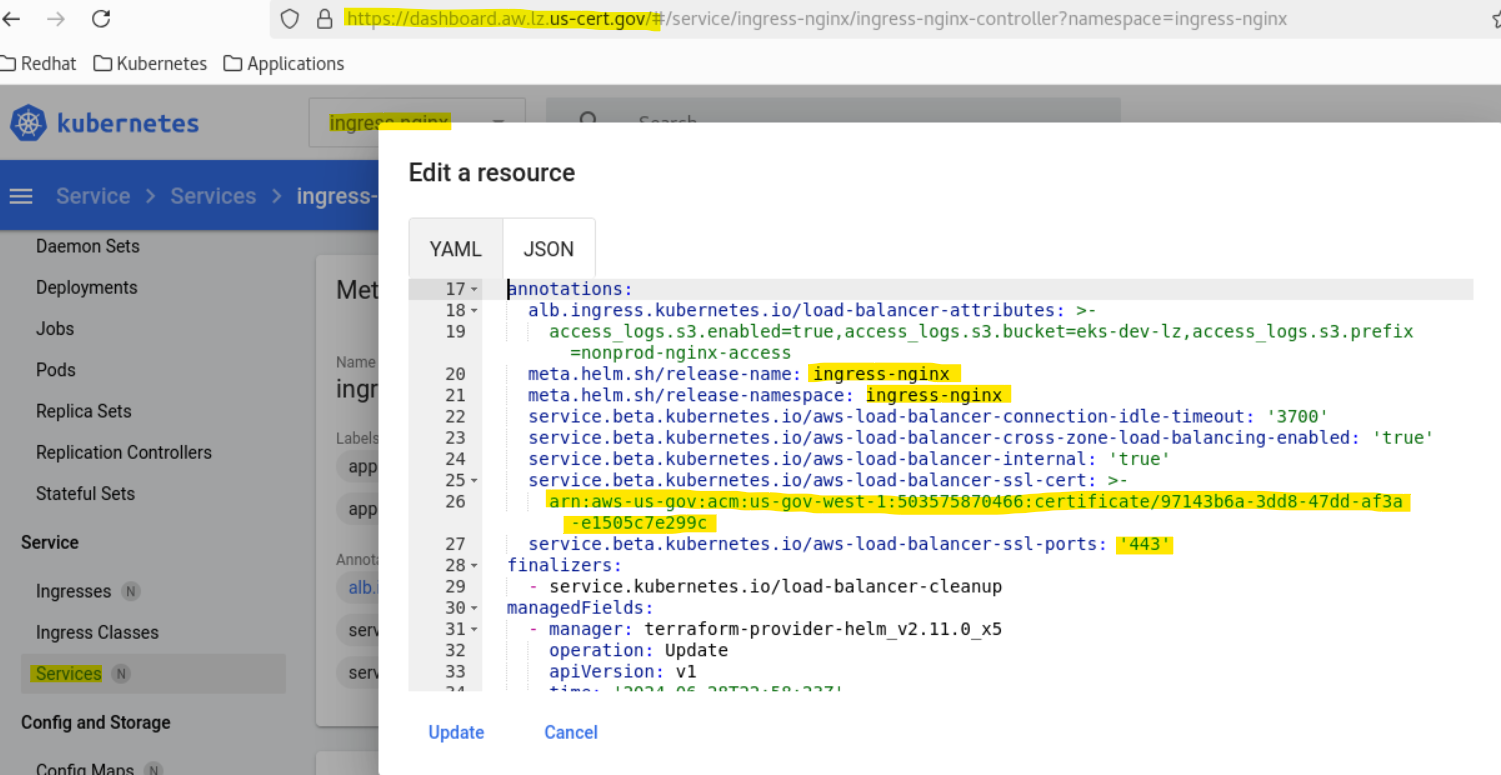


[**https://ui-uads-dev.aw.lz.us-cert.gov/**](https://ui-uads-dev.aw.lz.us-cert.gov/)



**Current Correct Value for Certificate:**

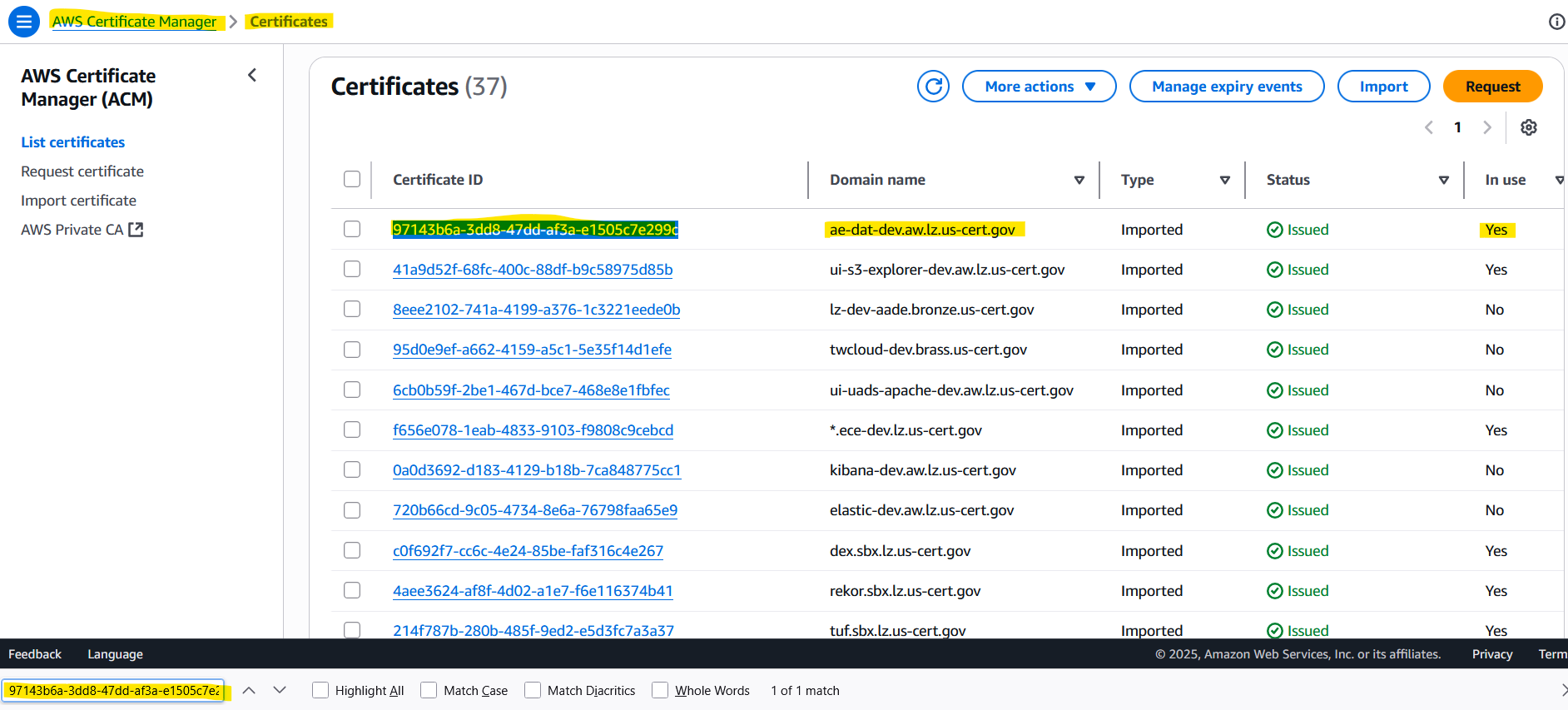
**Notice that, this is the correct certificate which include all the domains and ssl-ports is "443", which would make them accessible on HTTPS not HTTP.**



**Notice that the updated correct certificate exist in ACM, and that's why the domains are reachable Now on HTTPS.**

**Also, Notice it show as "In Use" = "Yes", after being loaded in the ingress-nginx controller manually.**

**After loading the correct certificate in the ingress-nginx, it will automatically restart the service but the pod will NOT restart.**



**All Domains are Reachable Now:**

**Notice that, all of these domains need to be reachable on HTTPS not HTTP from Redhat Workspaces.**

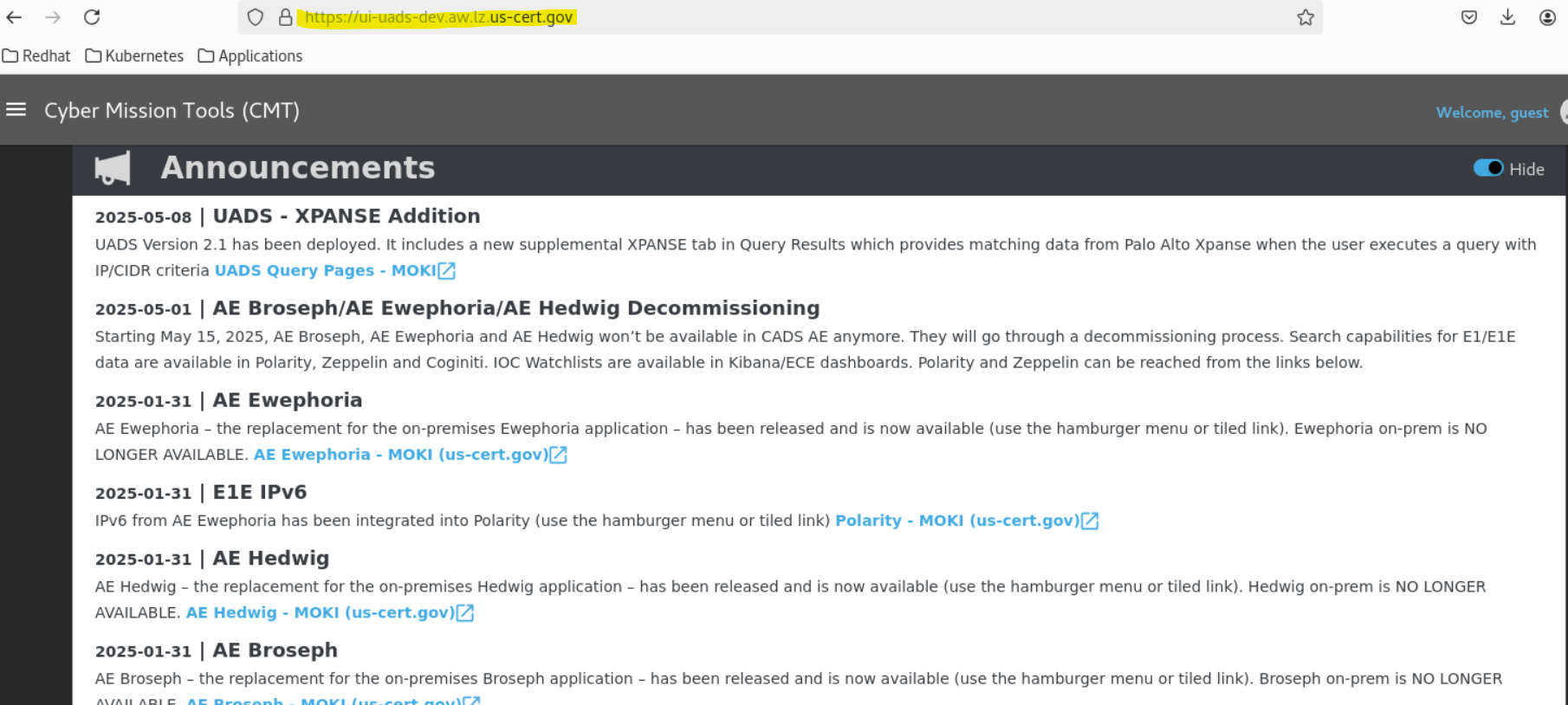
* [**https://ui-uads-dev.aw.lz.us-cert.gov/**](https://ui-uads-dev.aw.lz.us-cert.gov/)

* [**https://ui-uads-test.aw.lz.us-cert.gov/**](https://ui-uads-test.aw.lz.us-cert.gov/)

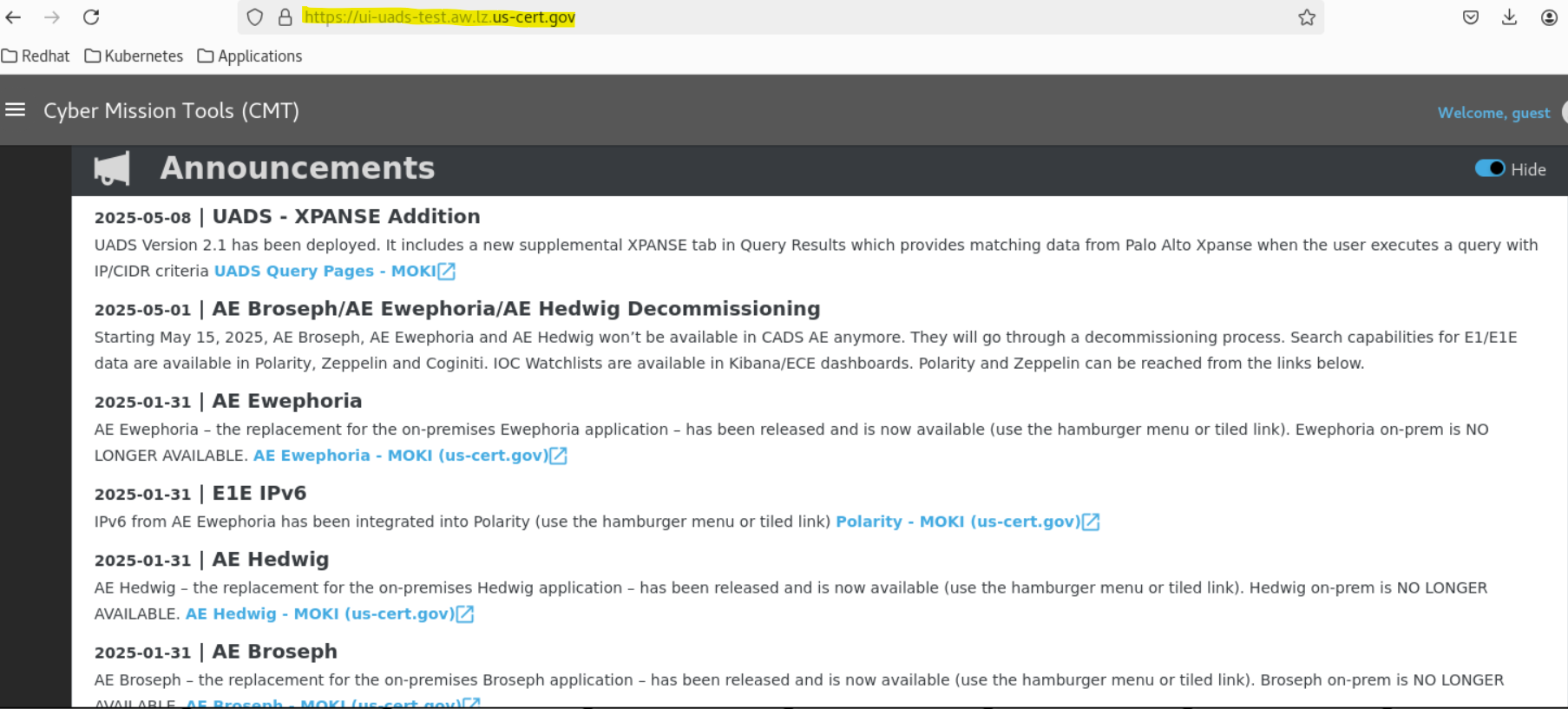
* [**https://ui-uads-team1-dev.aw.lz.us-cert.gov/**](https://ui-uads-team1-dev.aw.lz.us-cert.gov/)

* [**https://ui-hedwig-dev.aw.lz.us-cert.gov/**](https://ui-hedwig-dev.aw.lz.us-cert.gov/)

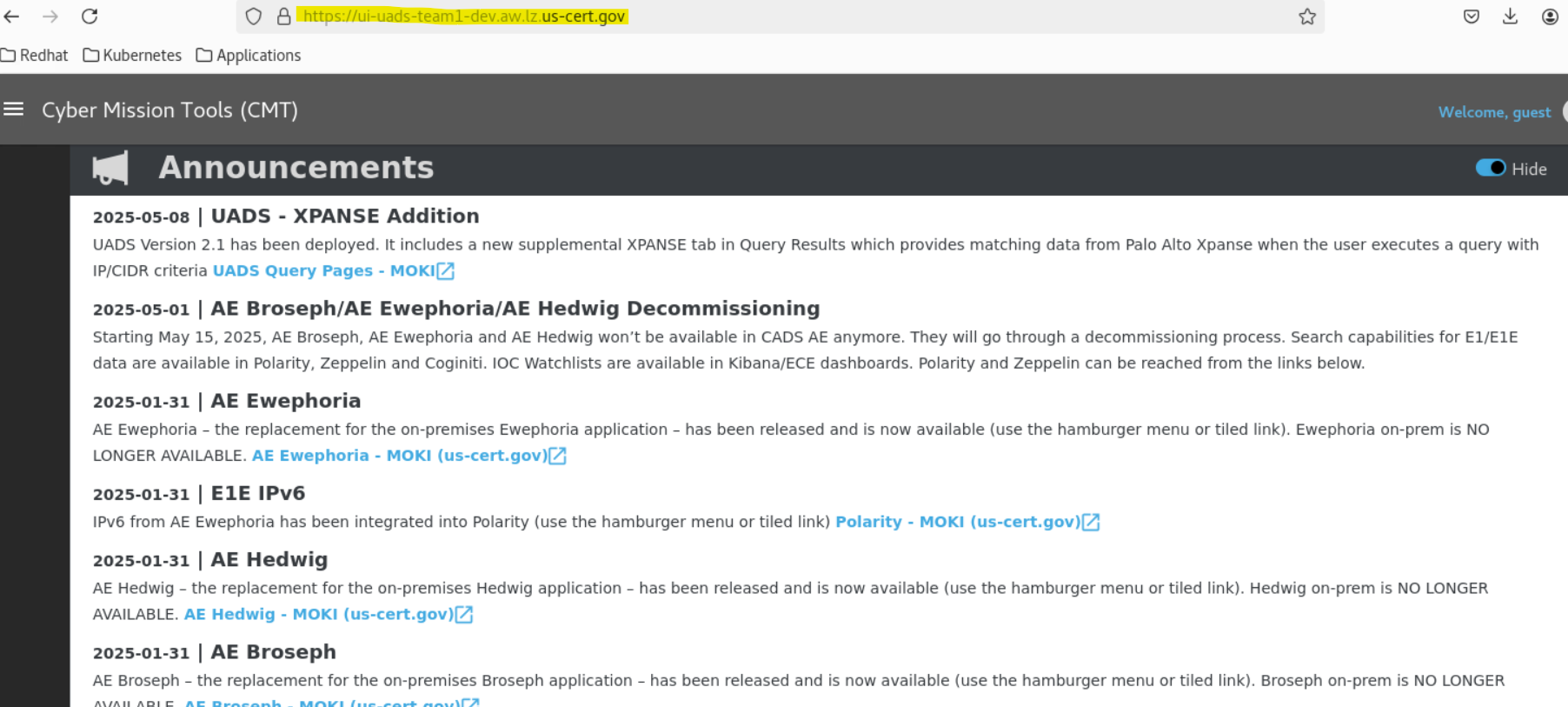
[**https://ui-uads-dev.aw.lz.us-cert.gov/**](https://ui-uads-dev.aw.lz.us-cert.gov/)



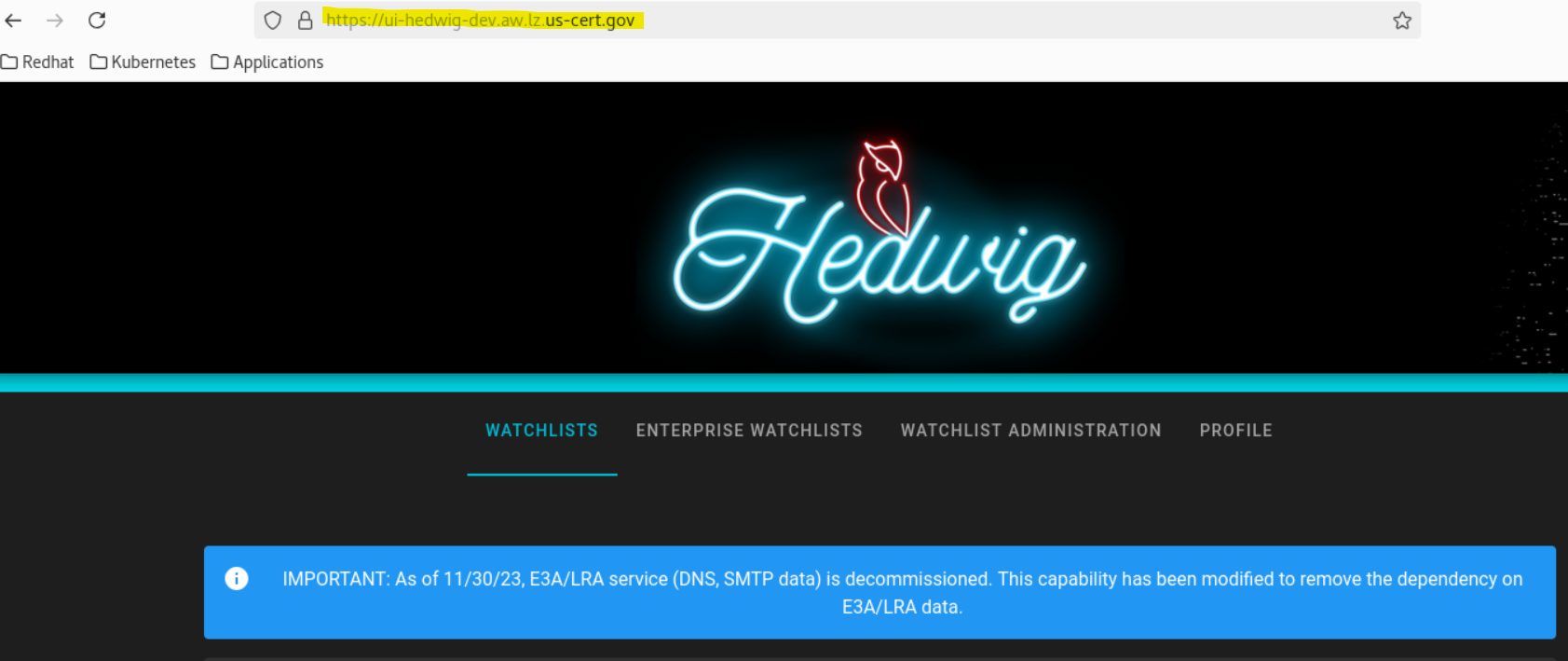
[**https://ui-uads-test.aw.lz.us-cert.gov/**](https://ui-uads-test.aw.lz.us-cert.gov/)



[**https://ui-uads-team1-dev.aw.lz.us-cert.gov/**](https://ui-uads-team1-dev.aw.lz.us-cert.gov/)



[**https://ui-hedwig-dev.aw.lz.us-cert.gov/**](https://ui-hedwig-dev.aw.lz.us-cert.gov/)



**VIP Notes:**

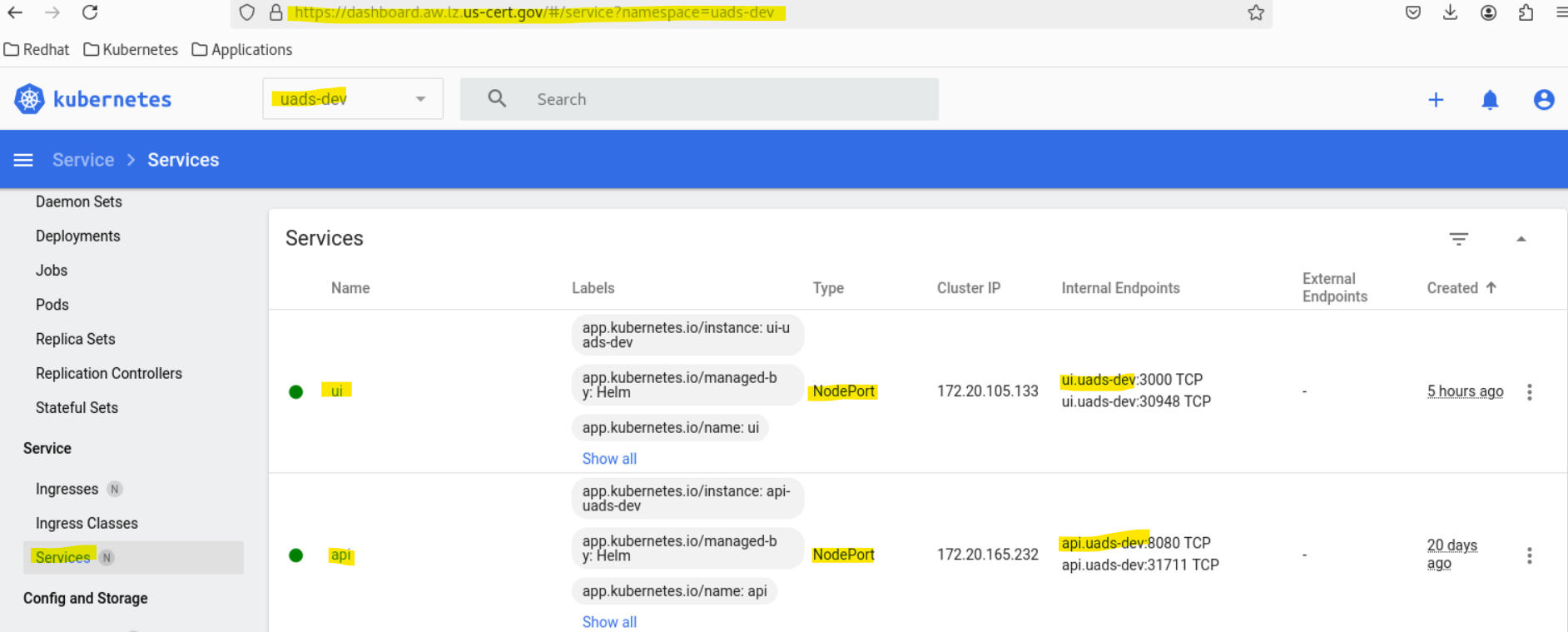
**---------------**

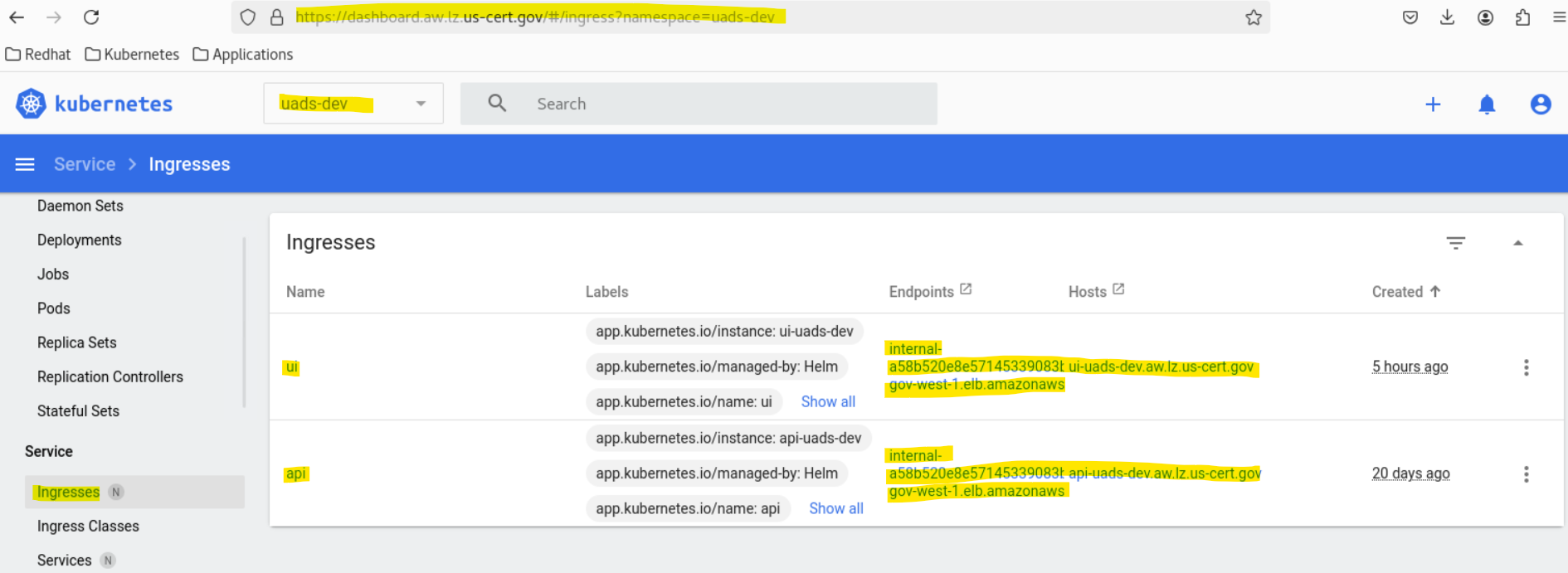
**All these domains share the same ALB Load Balancer, which was provisioned by the Nginx ingress controller.**

**Example:**

**Both Services for UI & API are being provisioned as NodePort, and they both share the same ALB Load Balancer which is provisioned by the Nginx ingress controller**

**Also, This ALB Load Balancer is the same Load Balancer in provisioned in the ingress-nginx namespace by the Nginx ingress controller.**

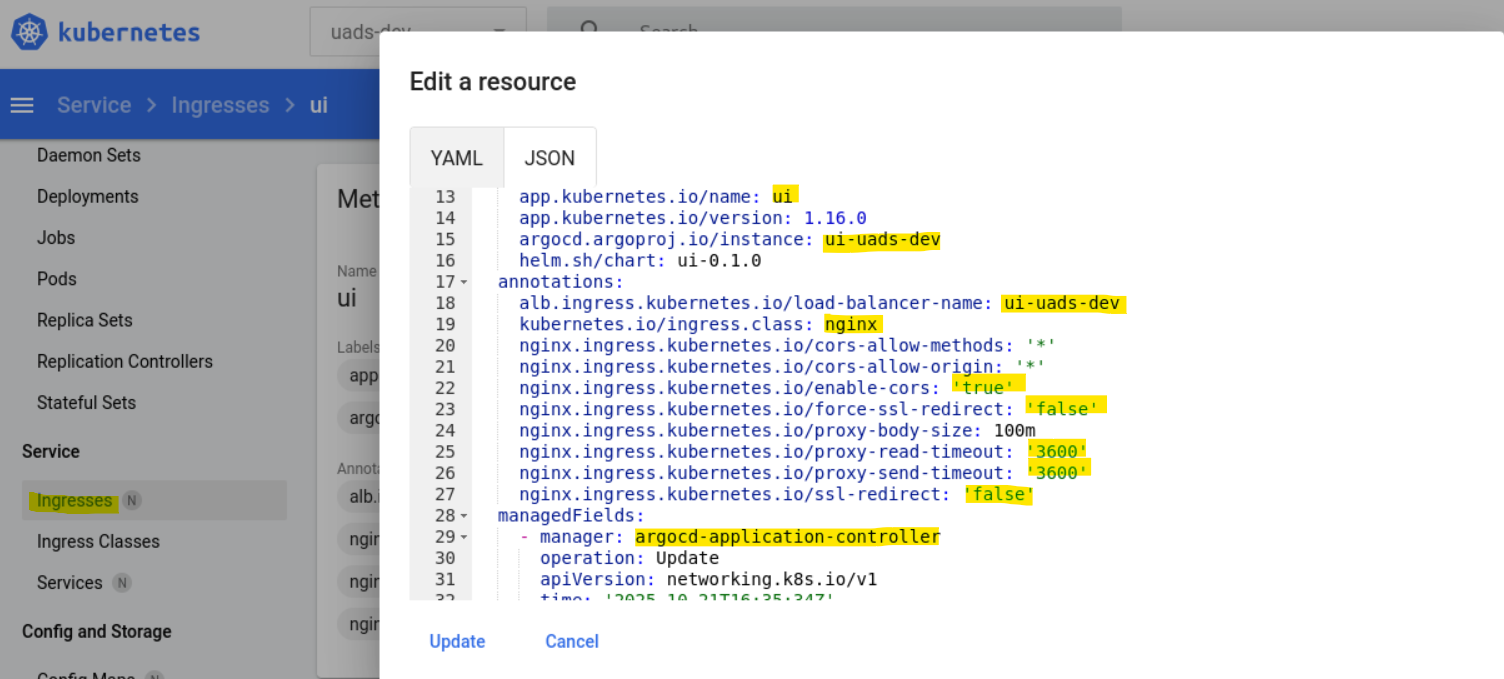


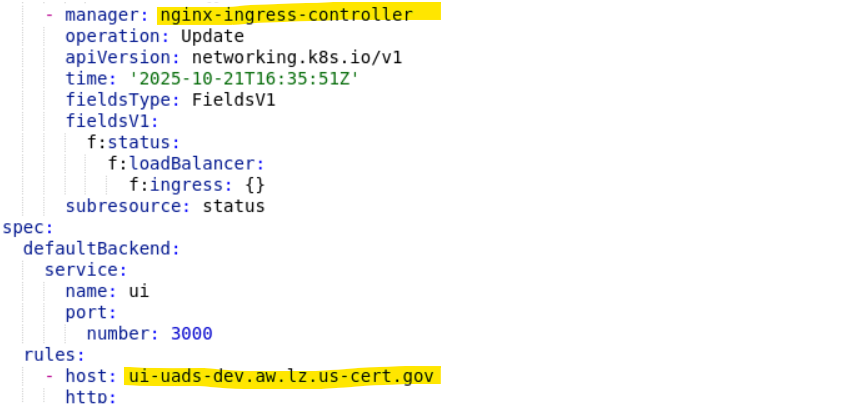


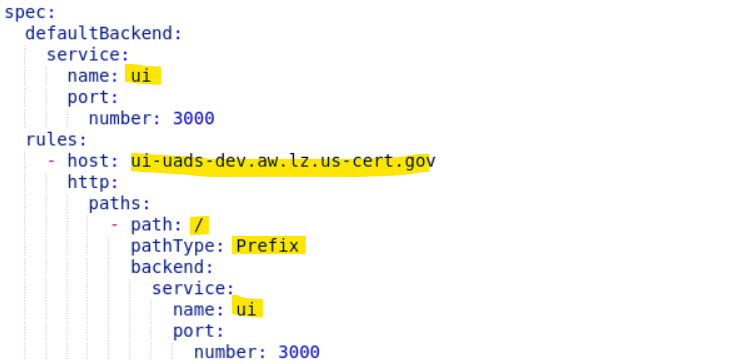
**UADS - UI-UADS-DEV - Service Example:**

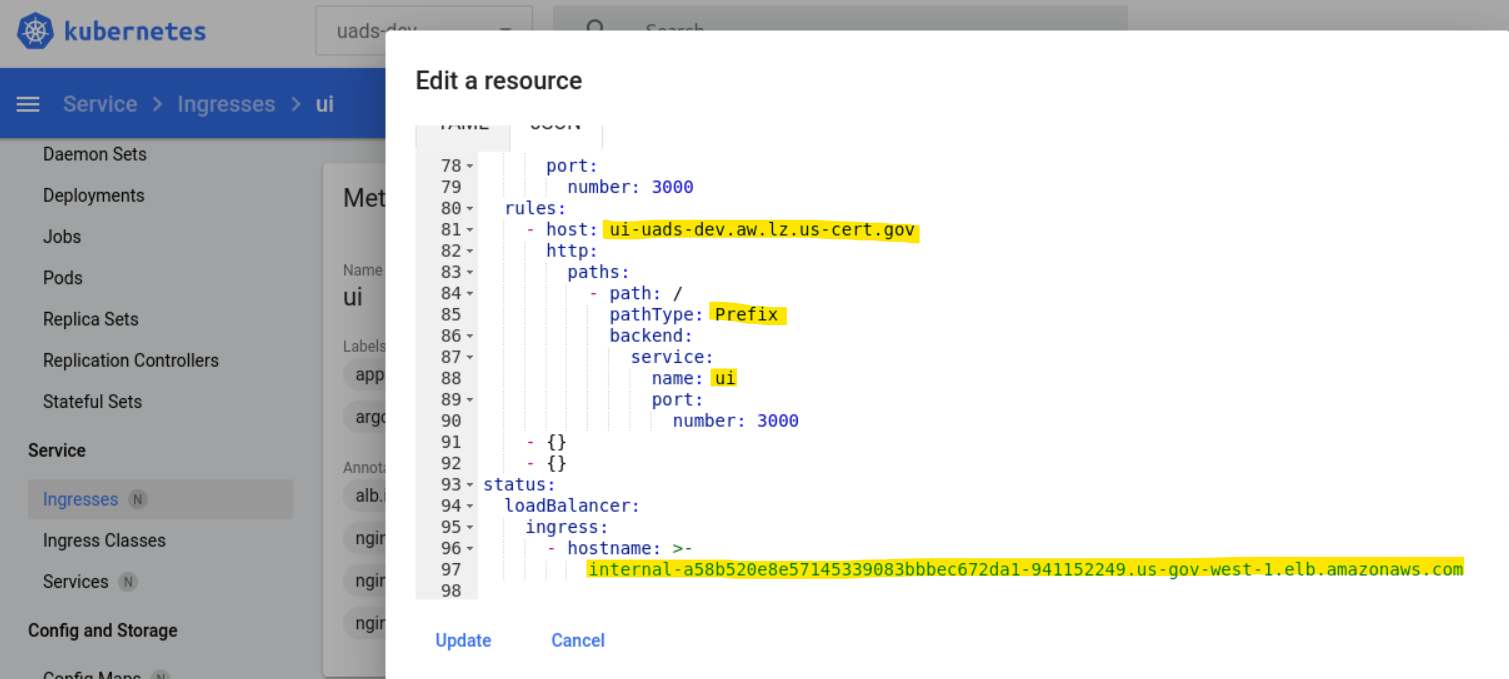
**Service Name:** [**https://ui-uads-dev.aw.lz.us-cert.gov/**](https://ui-uads-dev.aw.lz.us-cert.gov/)

**Load Balancer: internal-a58b520e8e57145339083bbbec672da1-941152249.us-gov-west-1.elb.amazonaws.com**









**kind: Ingress**

apiVersion: networking.k8s.io/v1

metadata:

**name: ui**

**namespace: uads-dev**

uid: 431c45c3-4f8c-4232-9a29-5b1ad0cbbfb2

resourceVersion: '314246827'

generation: 1

creationTimestamp: '2025-10-21T16:35:34Z'

**labels:**

**app.kubernetes.io/instance: ui-uads-dev**

app.kubernetes.io/managed-by: **Helm**

**app.kubernetes.io/name: ui**

app.kubernetes.io/version: 1.16.0

**argocd.argoproj.io/instance: ui-uads-dev**

helm.sh/chart: ui-0.1.0

**annotations:**

**alb.ingress.kubernetes.io/load-balancer-name: ui-uads-dev**

**kubernetes.io/ingress.class: nginx**

**nginx.ingress.kubernetes.io/cors-allow-methods: '\*'**

**nginx.ingress.kubernetes.io/cors-allow-origin: '\*'**

**nginx.ingress.kubernetes.io/enable-cors: 'true'**

**nginx.ingress.kubernetes.io/force-ssl-redirect: 'false'**

**nginx.ingress.kubernetes.io/proxy-body-size: 100m**

**nginx.ingress.kubernetes.io/proxy-read-timeout: '3600'**

**nginx.ingress.kubernetes.io/proxy-send-timeout: '3600'**

**nginx.ingress.kubernetes.io/ssl-redirect: 'false'**

managedFields:

**- manager: argocd-application-controller**

operation: Update

apiVersion: networking.k8s.io/v1

time: '2025-10-21T16:35:34Z'

fieldsType: FieldsV1

fieldsV1:

f:metadata:

f:annotations:

.: {}

f:alb.ingress.kubernetes.io/load-balancer-name: {}

f:kubernetes.io/ingress.class: {}

f:nginx.ingress.kubernetes.io/cors-allow-methods: {}

f:nginx.ingress.kubernetes.io/cors-allow-origin: {}

f:nginx.ingress.kubernetes.io/enable-cors: {}

f:nginx.ingress.kubernetes.io/force-ssl-redirect: {}

f:nginx.ingress.kubernetes.io/proxy-body-size: {}

f:nginx.ingress.kubernetes.io/proxy-read-timeout: {}

f:nginx.ingress.kubernetes.io/proxy-send-timeout: {}

f:nginx.ingress.kubernetes.io/ssl-redirect: {}

f:labels:

.: {}

f:app.kubernetes.io/instance: {}

f:app.kubernetes.io/managed-by: {}

f:app.kubernetes.io/name: {}

f:app.kubernetes.io/version: {}

f:argocd.argoproj.io/instance: {}

f:helm.sh/chart: {}

f:spec:

f:defaultBackend:

.: {}

f:service:

.: {}

f:name: {}

f:port: {}

f:rules: {}

**- manager: nginx-ingress-controller**

operation: Update

apiVersion: networking.k8s.io/v1

time: '2025-10-21T16:35:51Z'

fieldsType: FieldsV1

fieldsV1:

f:status:

f:loadBalancer:

f:ingress: {}

subresource: status

spec:

defaultBackend:

service:

**name: ui**

port:

number: 3000

rules:

**- host: ui-uads-dev.aw.lz.us-cert.gov**

http:

paths:

- path: /

pathType: Prefix

backend:

service:

**name: ui**

port:

number: 3000

- {}

- {}

status:

loadBalancer:

ingress:

- hostname: >-

**internal-a58b520e8e57145339083bbbec672da1-941152249.us-gov-west-1.elb.amazonaws.com**

**HELM Configurations of Ingress for UADS-UI**

**All the above configurations that you see in the NGINX ALB Ingress Load Balancer are being configured in this Helm chart.**

**Notice the following configurations:**

* **Service:** 
  + **type: NodePort**
* **Ingress:**
  + **enabled: true**
  + **kubernetes.io/ingress.class: nginx**

