**Instalación de subsistemas**

**NOTA:** El procedimiento detallado se creó el 17 de febrero del 2021. Debe consultarse las referencias detalladas en cada sección para confirmar si existen cambios en el procedimiento.

**Management**

**Referencia:** [**https://www.ibm.com/support/knowledgecenter/en/SSMNED\_v10/com.ibm.apic.install.doc/tapic\_v10\_install\_kubernetes\_mgmt.html**](https://www.ibm.com/support/knowledgecenter/en/SSMNED_v10/com.ibm.apic.install.doc/tapic_v10_install_kubernetes_mgmt.html)

1. Se debe definir y crear el storage previamente para poder seguir con la instalación de los subsistemas.
2. Editar el archivo management\_cr.yaml reemplazando:

$APP\_PRODUCT\_VERSION: 10.0.2.0

$SECRET\_NAME: apic-registry-secret

$PROFILE: n1xc4.m16

$DOCKER\_REGISTRY: 192.168.43.188:5000

$STACK\_HOST: hostname de la subred que se publicará en k8s, por ejemplo myhost.subnet.example.com

$STORAGE\_CLASS: el storage class para el almacenamiento persistente, por ejemplo “local-storage” o “myblock”

1. Editar el archivo management\_cr.yaml, dentro de “license” completar de la siguiente manera:

license:

accept: true

use: nonproduction

license: 'L-RJON-BZ5LJ5'

**Instalación en el DC Activo:**

1. Configure DC1 para que esté activo para el servicio Gestor de API en DC1.

Edite el archivo del CR ManagementCluster management\_cr de DC1 y configure las propiedades de la sección multiSiteHA, por ejemplo:

siteName: dallas

multiSiteHA:

mode: active

replicationEndpoint:

annotations:

certmanager.k8s.io/issuer: ingress-issuer

hosts:

- name: mgrreplicationdallas.cluster1.example.com

secretName: dc1-mgmt-replication

replicationPeerFQDN: mgrreplicationraleigh.cluster2.example.com

tlsClient:

secretName: dc1-mgmt-replication-client

1. Cree un secret denominado **mgmt-encryption-key** en los centros de datos activos y pasivos.

Debe crear un secret en los sitios activos y pasivos que utilice la misma cadena aleatoria. Ejecute el comando siguiente:

kubectl create secret generic mgmt-encryption-key --from-literal=encryption\_secret.bin=<RANDOM STRING> -n <namespace>

**Nota:** No debe utilizar la misma cadena aleatoria utilizada para crear el secret de **ptl-encryption-key**.

La cadena puede constar de letras mayúsculas, letras minúsculas y números, y debe tener un mínimo de 64 caracteres, pero no más de 100.

1. Cuando se hayan creado los secret en el DC1 y el DC2, actualice el Management CR en el DC1 y el DC2 para incluirlo en el objeto spec:

encryptionSecret:

secretName: mgmt-encryption-key

**Nota:** Asegúrese de añadir el secret al objeto spec y no al objeto multiSiteHA.

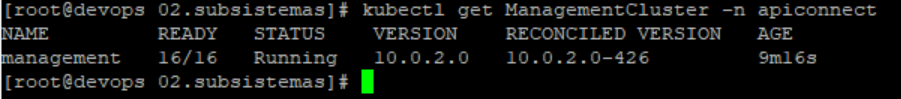
1. Instale el subsistema Management en el DC1

Por ejemplo, para aplicar el archivo del CR ManagementCluster al clúster DC1, ejecute este comando:

*kubectl apply -f management\_cr.yaml -n apiconnect*

1. Verificar que se encuentra instalado correctamente en el DC1 con el comando:

*kubectl get ManagementCluster -n apiconnect*



La instalación se ha completado cuando la columna READY es True y el SUMMARY informa que todos los servicios están en línea (por ejemplo, 14/14).

No es necesario esperar a que el estado LISTO sea Verdadero antes de pasar a la siguiente instalación del Subsistema.

1. Configure el direccionador dinámico para que dirija todo el tráfico a DC1.

Esto incluye la definición de los cuatro endpoint para el Manager API clúster.

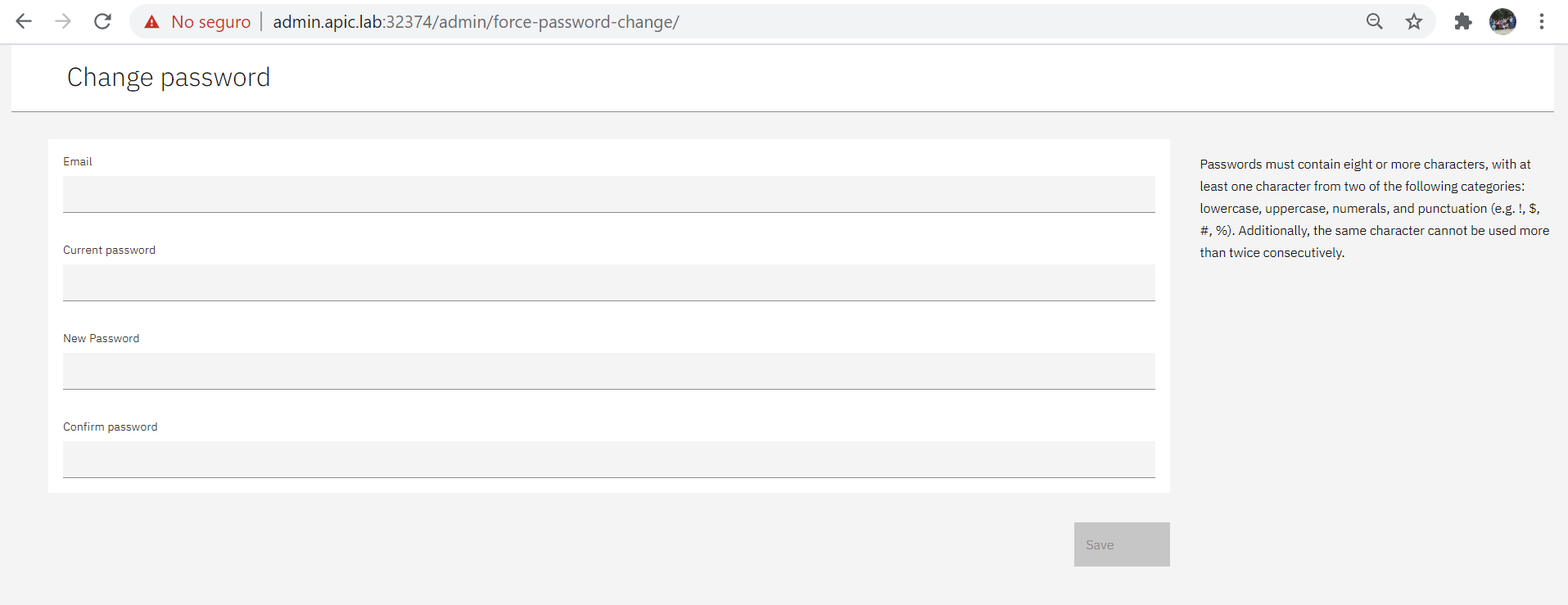
1. Verifique su conexión a la interfaz de usuario de Cloud Manager en el subsistema Management en su endpoint Cloud Manager.

https: //admin.YOUR\_HOSTNAME/admin

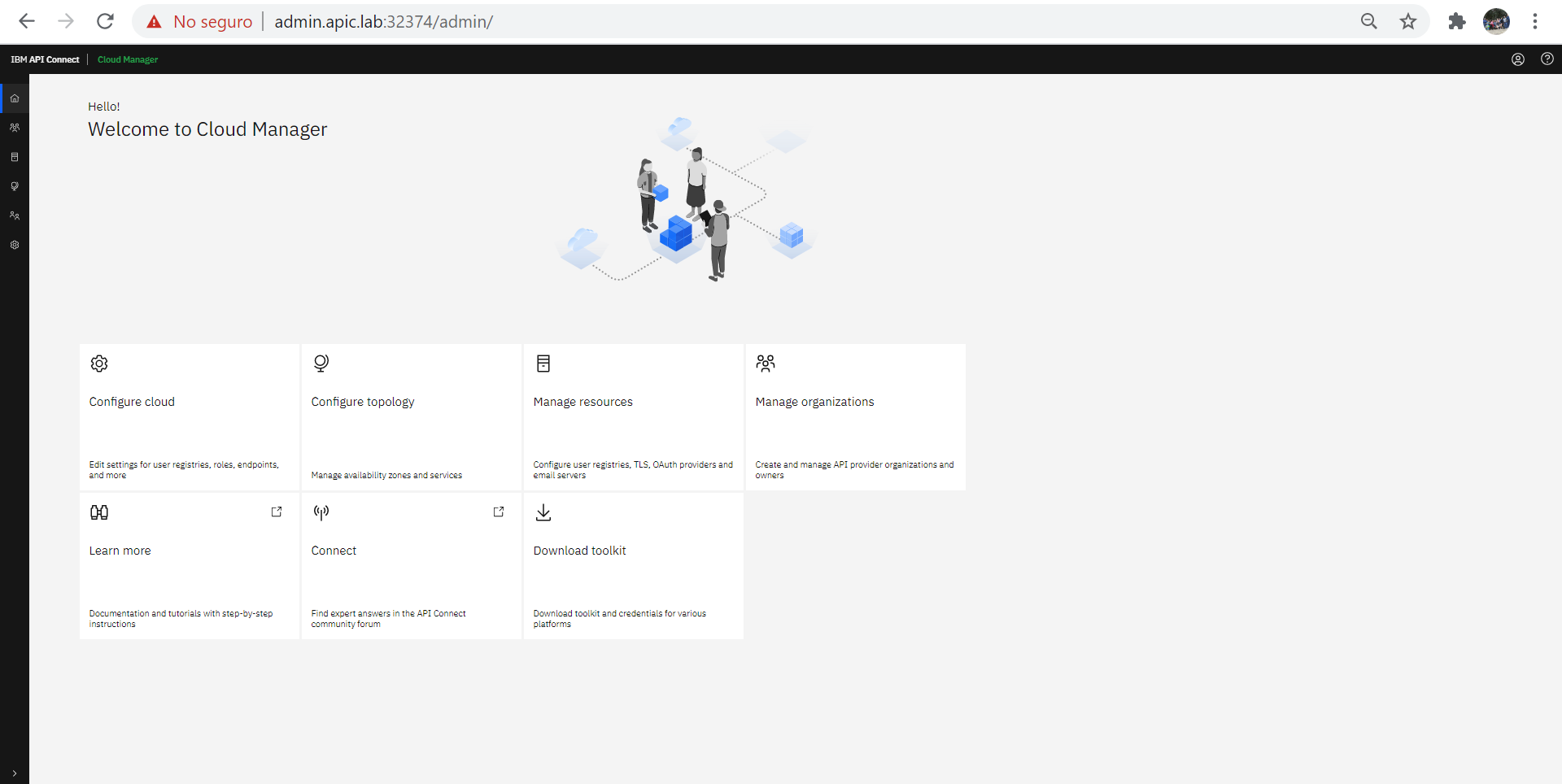
https://admin.apic.lab/admin

Acceda con las credenciales: admin / 7iron-hide

Se le solicitará cambiar las credenciales:



Ingresará al home de Cloud Manager:



**Instalación en el DC Pasivo:**

**OBS:** Tiene que utilizar la misma clave de cifrado en ambos sitios

1. Configure DC2 para que sea pasivo para el API Manager service en DC2.

multiSiteHA:

mode: passive

replicationEndpoint:

annotations:

certmanager.k8s.io/issuer: ingress-issuer

hosts:

- name: mgrreplicationraleigh.cluster2.example.com

secretName: raleigh-mgr-replication-worker-1

replicationPeerFQDN: mgrreplicationdallas.cluster1.example.com

tlsClient:

secretName: mgr-replication-client

...

siteName: raleigh

1. Asegúrese de haber completado los pasos del apartado Instalación en un centro de datos activo para crear un secret denominado mgmt-encryption-key en el DC1 y el DC2, así como para actualizar el Management CR en el DC1 y el DC2 a fin de incluir el secret mgmt-encryption-key en el objeto spec.
2. Instale el subsistema Management en el DC2

Por ejemplo, para aplicar el archivo del CR ManagementCluster al clúster DC2, ejecute este comando:

*kubectl apply -f management\_cr.yaml -n apiconnect*

1. Verificar que se encuentra instalado correctamente en el DC1 con el comando:

*kubectl get ManagementCluster -n apiconnect*

**Developer Portal**

**Referencia:** <https://www.ibm.com/support/knowledgecenter/en/SSMNED_v10/com.ibm.apic.install.doc/tapic_v10_install_kubernetes_portal.html>

1. Editar el archivo portal\_cr.yaml reemplazando:

$APP\_PRODUCT\_VERSION: 10.0.2.0

$PROFILE:n1xc2.m8

$SECRET\_NAME:apic-registry-secret

$DOCKER\_REGISTRY: nombre del registro de docker

$STACK\_HOST: hostname de la subred que se publicará en k8s, por ejemplo myhost.subnet.example.com

$STORAGE\_CLASS: el storage class para el almacenamiento persistente, por ejemplo “local-storage”

1. Editar el archivo portal\_cr.yaml, dentro de “license” completar de la siguiente manera:

license:

accept: true

use: nonproduction

license: 'L-RJON-BZ5LJ5'

1. Editar el archivo portal\_cr.yaml, dentro de “portalAdminEndpoint” modificar el valor de “name” para asociarlo a la capa de exposición respectiva, por ejemplo:

name: api.portal.apiconnect.myhost.subnet.example.com

name: api.portal.apiconnect.apic.lab

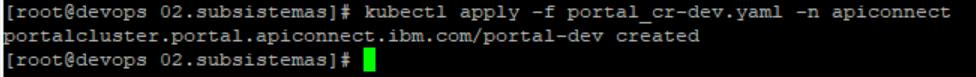
1. Editar el archivo portal\_cr.yaml, dentro de “portalUIEndpoint” modificar el valor de “name” para asociarlo a la capa de exposición respectiva, por ejemplo:

name: portal.apiconnect.myhost.subnet.example.com

name: portal.apiconnect.apic.lab

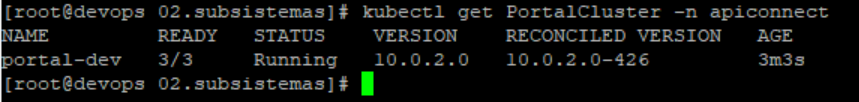
1. Instalar el subsistema con el commando:

*kubectl apply -f portal\_cr.yaml -n apiconnect*



1. Verificar que se encuentra instalado con el comando:

*kubectl get PortalCluster -n apiconnect*



1. Respaldar la semilla de encriptación con los siguientes comandos:

*kubectl get ptl portal -n <namespace> -o yaml | grep encryptionSecret*

*kubectl get secret portal-enc-key -n <namespace> -o yaml > portal-enc-key.yaml*

Guarde el archivo *portal-enc-key.yaml* en un lugar seguro.



**Gateway**

**Referencia:** <https://www.ibm.com/support/knowledgecenter/en/SSMNED_v10/com.ibm.apic.install.doc/tapic_v10_install_kubernetes_gwy.html>

1. Editar el archivo apigateway\_cr.yaml reemplazando:

$APP\_PRODUCT\_VERSION: 10.0.1.1-eus

$PROFILE:n1xc4.m8

$SECRET\_NAME:apic-registry-secret

$DOCKER\_REGISTRY: nombre del registro de docker

$STACK\_HOST: hostname de la subred que se publicará en k8s, por ejemplo myhost.subnet.example.com

$STORAGE\_CLASS: el storage class para el almacenamiento persistente, por ejemplo “local-storage”

$ADMIN\_USER\_SECRET: datapower-admin-credentials

1. Editar el archivo apigateway\_cr.yaml, dentro de “license” completar de la siguiente manera:

license:

accept: true

use: nonproduction

1. Editar el archivo apigateway\_cr.yaml, dentro de “gatewayEndpoint” modificar el valor de “name” para asociarlo al namespace, por ejemplo:

name: rgw.intranet.myhost.subnet.example.com

1. Editar el archivo apigateway\_cr.yaml, dentro de “gatewayManagerEndpoint” modificar el valor de “name” para asociarlo al namespace, por ejemplo:

name: rgwd.intranet.myhost.subnet.example.com

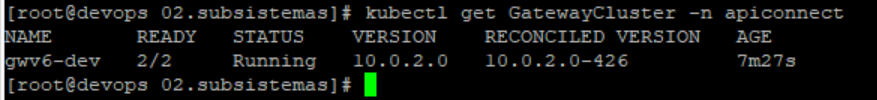
1. Instalar el subsistema con el commando:

*kubectl apply -f apigateway\_cr.yaml -n apiconnect*



1. Verificar que se encuentra instalado con el comando:

*kubectl get GatewayCluster -n apiconnect*



**Analytics**

**Referencia:** <https://www.ibm.com/support/knowledgecenter/en/SSMNED_v10/com.ibm.apic.install.doc/analytics_install_create_cr_k8s.html>

1. Aumentar el número de mapas en los workers:

*sudo sysctl -w vm.max\_map\_count=262144*

1. Persistir el cambio en los workers editando el archivo */etc/sysctl.conf* agregando la siguiente línea:

*vm.max\_map\_count = 262144*

1. Editar el archivo analytics\_cr.yaml reemplazando:

$APP\_PRODUCT\_VERSION: 10.0.1.1-eus

$PROFILE:n1xc2.m16

$SECRET\_NAME:apic-registry-secret

$DOCKER\_REGISTRY: nombre del registro de docker

$STACK\_HOST: hostname de la subred que se publicará en k8s, por ejemplo myhost.subnet.example.com

$STORAGE\_CLASS: el storage class para el almacenamiento persistente, por ejemplo “local-storage”

$DATA\_VOLUME\_SIZE: colocar el tamaño de data a almacenar

$VOLUME\_SIZE: colocar el tamaño de data maestra, para no productivos 5Gi para productivo 10Gi

1. Editar el archivo analytics\_cr.yaml, dentro de “license” completar de la siguiente manera:

license:

accept: true

use: nonproduction

1. Editar el archivo analytics\_cr.yaml, dentro de “client” modificar el valor de “name” para asociarlo al namespace, por ejemplo:

endpoint:

hosts:

- name: ac.intranet.myhost.subnet.example.com

1. Editar el archivo analytics\_cr.yaml, dentro de “gatewayManagerEndpoint” modificar el valor de “name” para asociarlo al namespace, por ejemplo:

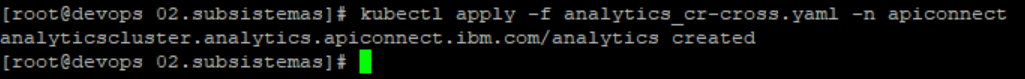
ingestion:

hosts:

- name: ai.intranet.myhost.subnet.example.com

1. Instalar el subsistema con el commando:

*kubectl apply -f analytics\_cr.yaml -n apiconnect*



1. Verificar que se encuentra instalado con el comando:

*kubectl get AnalyticsCluster -n apiconnect*

