LABORATORIOS DE ARQUITECTURA EN ORACLE CLOUD INFRASTRUCTURE

Manual de consola — Seguridad cloud de extremo a extremo

Red privada en tres capas, NSG por función, WAF sobre el balanceador, bastión, flow logs, Cloud Guard y Security Zone, construidos a mano

Módulo 04

Primera pasada 3 h la primera pasada por consola

Costo aproximado unidades de USD; el ensayo midió 0,13 USD de cómputo y el presupuesto proyectó 2,10 USD

Vía consola de Oracle Cloud, pantalla por pantalla

Este documento reproduce desde la consola lo que el laboratorio automatiza con Terraform. Los dos caminos llegan al mismo sitio: el código sirve para repetirlo, la consola para entenderlo.

El laboratorio corre sobre un tenancy de prueba desechable. Las decisiones de acceso que aquí se toman no son una referencia de configuración segura para un ambiente con datos; cuando alguna se aparta de la buena práctica, el manual lo dice en el sitio donde se aparta.

Este manual construye desde la consola, haciendo clic, lo mismo que el módulo de Terraform de este laboratorio crea en un `apply`. No hace falta Terraform ni la CLI de OCI para seguirlo: la CLI aparece solo en los bloques de verificación, donde una línea confirma en un segundo lo que en la consola tomaría tres pantallas. El laboratorio se despliega en un tenancy de prueba, desechable; no se toca ningún ambiente con datos reales.

1. Qué se construye y para qué

Se construyen **dos laboratorios** y se auditan con **el mismo script**.

El primero está hecho a la ligera: una sola subred pública, servidores de aplicación con IP pública, la security list por defecto tal como OCI la entrega, un balanceador sin filtro de aplicación y ningún registro de tráfico. No es una caricatura: es cómo queda un ambiente que se levantó para salir del paso y nadie volvió a mirar.

El segundo está hecho bien: tres capas de red, nada con IP pública salvo el balanceador, reglas de tráfico por función, un firewall de aplicación en el borde, acceso administrativo por bastión con sesiones que expiran, registro de flujos de red, alertas ante cambios críticos y detección continua de postura.

La decisión que el laboratorio ayuda a tomar no es si se compra seguridad, sino **cuál es el mínimo de controles que un ambiente debe cumplir antes de recibir tráfico real, y quién responde por cada uno**. La auditoría pone números sobre la mesa para que esa conversación no sea de opiniones.

El contraste, medido

```
A · construido a la ligera              B · construido bien
-----------------------------------     -----------------------------
una subred, publica                     tres subredes, una publica
dos instancias con IP publica           instancias sin IP publica
SL por defecto: 22 desde 0.0.0.0/0      SL por defecto endurecida
balanceador sin WAF                     balanceador con WAF
sin flow logs                           flow logs en la subred de app
acceso SSH directo desde internet       bastion con TTL y lista de CIDR
-----------------------------------     -----------------------------
5 hallazgos altos                       0 hallazgos altos
3 medios · 2 controles que cumplen      2 medios · 6 que cumplen
```

Los cinco hallazgos altos del laboratorio A son: balanceador público sin WAF, dos instancias con IP pública y dos security lists que abren el puerto 22 a `0.0.0.0/0` —una de ellas, la que OCI crea sola con cada VCN—. Cloud Guard aparece como sexto hallazgo alto cuando está apagado; en la corrida medida ya estaba encendido.

Arquitectura del laboratorio B

```
                        Internet
                           | 80 / 443
      +--------------------v--------------------+
```

```
        |  Load Balancer publico  +  WAF          |  subred publica
        |  NSG-lb: entra 80 y 443 desde internet  |  10.40.1.0/24
        +--------------------+--------------------+
                             | 80  (solo desde NSG-lb)
        +--------------------v--------------------+
        |  app-1      app-2      sin IP publica   |  subred privada app
        |  NSG-app: 80 desde NSG-lb               |  10.40.2.0/24
        |           22 desde el endpoint bastion  |  salida: NAT + SGW
        +---+-----------------------------+-------+
            | 3306 (solo desde NSG-app)   | 22
        +---v---------------+    +---------+---------+
        |  subred privada   |    |  OCI Bastion      |
        |  datos            |    |  sesion con TTL   |
        |  10.40.3.0/24     |    |  lista de CIDR    |
        |  sin ruta a       |    +-------------------+
        |  internet         |
        +-------------------+

    Flow logs en la subred de app · Events + Notifications ante cambios de
    red e IAM · Cloud Guard sobre el tenancy · Security Zone en su propio
    compartimento
```

La capa de datos se construye aunque este laboratorio no despliegue ninguna base de datos: así el diagrama de tres capas se puede mostrar completo en la consola, y la regla «el puerto de la base solo se alcanza desde la aplicación» existe como objeto y no como intención.

2. Equivalencias con otras nubes

Solo los servicios que aparecen en este manual.

Concepto	AWS	Azure	OCI
Frontera de facturación y permisos	Cuenta / OU	Suscripción / Grupo de recursos	**Compartimento**
Red virtual	VPC	Virtual Network	**VCN**
Subred privada	Subnet privada	Subnet sin IP pública	**Subred con IP pública prohibida**
Salida a internet sin entrada	NAT Gateway	NAT Gateway	**NAT Gateway**
Acceso privado a servicios del proveedor	VPC Endpoint / PrivateLink	Private Endpoint / Service Endpoint	**Service Gateway**
Filtro a nivel de subred	Network ACL	(no hay equivalente directo)	**Security list**
Filtro a nivel de interfaz	Security Group	Network Security Group	**Network Security Group (NSG)**
Balanceador de capa 7	Application Load Balancer	Application Gateway	**Load Balancer (flexible)**
Firewall de aplicación	AWS WAF	Azure WAF (sobre App Gateway o Front Door)	**OCI WAF**

Acceso administrativo sin exponer SSH	Systems Manager Session Manager / EC2 Instance Connect Endpoint	Azure Bastion	**OCI Bastion**
Registro de flujos de red	VPC Flow Logs	NSG Flow Logs	**Flow logs (servicio Logging)**
Registro de llamadas al plano de control	CloudTrail	Activity Log	**Audit**
Postura de seguridad continua	Security Hub / GuardDuty	Microsoft Defender for Cloud	**Cloud Guard**
Barrera que impide crear lo que viola la política	SCP (parcial) / Config Rules con remediación	Azure Policy en modo Deny	**Security Zone**
Eventos del plano de control → acción	EventBridge	Event Grid	**Events Service**
Notificación por correo	SNS	Action Group	**Notifications**

Dos diferencias que sorprenden a quien llega de AWS o Azure:

- **Un NSG se referencia a otro como origen o destino.** Una regla puede decir literalmente "acepto el puerto 80 cuando el origen es el NSG del balanceador", sin escribir CIDR ni mantener listas de IP. Es la razón por la que aquí las security lists quedan casi vacías.
- **Los compartimentos no son grupos de recursos.** Son la unidad sobre la que se escriben las políticas de IAM, se mide el costo y se aplican las Security Zones.

3. Prerrequisitos

3.1 Permisos

Quien siga este manual necesita, sobre el compartimento del laboratorio, permisos para crear red, cómputo, balanceadores, WAF, bastión y logs; y sobre el **tenancy**, permiso para habilitar Cloud Guard y para crear la Security Zone y su receta. Si es administrador del tenancy de prueba, ya los tiene.

Si prefiere un usuario acotado, el conjunto mínimo se ve así:

```
Allow group LabSeguridad to manage virtual-network-family in compartment lab
Allow group LabSeguridad to manage instance-family in compartment lab
Allow group LabSeguridad to manage load-balancers in compartment lab
Allow group LabSeguridad to manage waf-family in compartment lab
Allow group LabSeguridad to manage bastion-family in compartment lab
Allow group LabSeguridad to manage logging-family in compartment lab
Allow group LabSeguridad to manage ons-family in compartment lab
Allow group LabSeguridad to manage cloudevents-rules in compartment lab
Allow group LabSeguridad to manage cloud-guard-family in tenancy
Allow group LabSeguridad to manage security-zone in tenancy
```

Las familias de recursos exactas cambian entre versiones del servicio; si la consola rechaza una operación, el mensaje nombra el verbo y el tipo que faltan `[VALIDAR]`.

3.2 Qué debe existir antes de empezar

Elemento	Cómo obtenerlo
Tenancy de prueba, vacío	Cuenta de prueba propia. No se usa un tenancy con datos
Región home definida	La región de reporte de Cloud Guard **no se puede cambiar después**
Un par de llaves SSH	`ssh-keygen -t rsa -b 4096`. Solo se usa a través del bastión
Un buzón de correo para alertas	Cualquiera al que tenga acceso. Hay que confirmar la suscripción
Compartimentos del laboratorio	Se crean en el capítulo 4

3.3 Límites de una cuenta de prueba

Antes de empezar conviene mirar los límites, porque el error de límite aparece a la mitad de la construcción y obliga a devolverse.

Consola de OCI Governance & Administration › Limits, Quotas and Usage

Recurso	Cuánto usa este laboratorio
Instancias `VM.Standard.E4.Flex`	2 (1 OCPU · 8 GB cada una)
Load balancers flexibles	1 (más 1 si también levanta el laboratorio A)
VCN	1 (más 1 para el laboratorio A)
Bastiones	1
Políticas de WAF	1

Si el límite de balanceadores es 1, construya un laboratorio a la vez y audite cada uno por separado. El contraste se pierde en simultaneidad, no en contenido: las dos salidas del script quedan guardadas y se comparan después.

3.4 Valores usados en todo el manual

Campo	Valor
Prefijo de nombres	`lab02-`
CIDR de la VCN	`10.40.0.0/16`
Subred pública (balanceador)	`10.40.1.0/24`
Subred privada de aplicación	`10.40.2.0/24`
Subred privada de datos	`10.40.3.0/24`

Imagen	Oracle Linux 9, la más reciente
Shape	VM.Standard.E4.Flex · 1 OCPU · 8 GB

Los OCID que aparecen en los ejemplos son ficticios y se ven como `ocid1.compartment.oc1..aaaaEJEMPLO`.

4. Compartimentos del laboratorio

Todo cuelga de un compartimento padre para que el presupuesto, el filtro de costos y el borrado final cubran los dos laboratorios de una vez.

```
lab                     <- el presupuesto apunta aqui
|- lab-01-elasticidad          <- laboratorio A, el construido a la ligera
|- lab-02-seguridad          <- laboratorio B, el construido bien
`- lab-02-zona-segura     <- vacio; se le asocia la Security Zone
```

Consola de OCI Identity & Security › Compartments › Create Compartment

1. Cree el padre.

Campo	Valor
Name	`lab`
Description	Laboratorios de seguridad
Parent Compartment	el tenancy raíz

1. En **Tags**, agregue dos etiquetas libres. Sirven para filtrar el costo después y para reconocer lo desechable de un vistazo.

Campo	Valor
Tag Key	`Proyecto`
Value	`Laboratorio`
Tag Key	`Efimero`
Value	`si`

1. Repita **Create Compartment** tres veces más, con `lab` como *Parent Compartment*, para crear `lab-01-elasticidad`, `lab-02-seguridad` y `lab-02-zona-segura`.
2. Anote los cuatro OCID. Los va a necesitar en la auditoría y en la Security Zone.

Un compartimento recién creado tarda un par de minutos en propagarse. Si aparece en la lista pero un formulario no lo ofrece todavía, recargue la página y espere dos minutos antes de

concluir que algo falló.

5. Cloud Guard

Cloud Guard va **primero**, antes de construir nada. No porque la construcción lo necesite, sino porque evalúa los recursos cuando cambian y en barridos periódicos: si se habilita al final, la consola sale vacía justo cuando se quiere mostrar algo. Habilitado antes, cuando el laboratorio A lleve unas horas arriba ya habrá problemas detectados sobre él.

Consola de OCI Identity & Security › Cloud Guard

1. Abra Cloud Guard. Si el servicio nunca se ha habilitado en este tenancy, la consola presenta directamente el asistente de habilitación.
2. Complete el asistente.

Campo	Valor
Reporting region	La región home del tenancy
Compartments to monitor	El tenancy raíz
Configuration detector recipe	La receta administrada por Oracle para configuración
Activity detector recipe	La receta administrada por Oracle para actividad

1. El asistente muestra que va a crear las **políticas de servicio** que Cloud Guard necesita para leer los recursos del tenancy. Acéptelas. Sin ellas el servicio queda habilitado pero ciego.
2. Confirme y espere. La habilitación tarda poco; el primer barrido completo, más.

IMPORTANTE · Cloud Guard no se puede habilitar sin crear antes las políticas de servicio

Por consola esto es invisible: el asistente redacta las sentencias, las muestra y las crea al aceptar. Es el camino fácil y es el que documenta este manual.

Por CLI no hay asistente. `oci cloud-guard configuration update` falla si las políticas de servicio no existen todavía, y el mensaje de error no dice que falten políticas: dice que la operación no está autorizada. En el ensayo de este laboratorio hicieron falta **17 sentencias de política** creadas a mano antes de que la habilitación por CLI funcionara. Todas tienen la forma `Allow service cloudguard to <verbo> <tipo de recurso> in tenancy`, y el conjunto exacto depende de las recetas de detección que se activen: la lista autoritativa es la que el propio asistente de la consola muestra en el momento de aceptar `[VALIDAR la lista contra la consola de su región]`.

Consecuencia práctica: **habilite Cloud Guard por consola una vez, y automatice todo lo demás.** No vale la pena scriptear la única operación del laboratorio que la consola resuelve sola.

Verificación:

```
oci cloud-guard configuration get --compartment-id "<ocid-tenancy>" \
  --query 'data.status'
# -> "ENABLED"
```

5.1 Dónde se ven los hallazgos

Consola de OCI Identity & Security › Cloud Guard › Problems

Deje pasar el tiempo. Cuando el laboratorio A esté arriba, esta pantalla debería mostrar problemas sobre sus recursos: instancias con IP pública, puerto de administración abierto, balanceador expuesto. Filtre por compartimento para separar el ruido del resto del tenancy. Cloud Guard **detecta**; no impide, y esa distinción es la que hace falta para entender el capítulo 17.

6. El laboratorio A — el construido a la ligera

Este laboratorio existe para ser auditado. Se construye rápido, a propósito, con las decisiones que se toman cuando el objetivo es que funcione hoy. Si ya tiene desplegado el laboratorio de elasticidad del módulo anterior, **úselo y salte este capítulo**: es exactamente este ambiente. Si no, lo que sigue reproduce lo mínimo para que la auditoría tenga qué encontrar.

Consola de OCI Networking › Virtual Cloud Networks › Create VCN

1. Cree la VCN.

Campo	Valor
Name	`lab01-vcn`
Compartment	`lab-01-elasticidad`
IPv4 CIDR Blocks	`10.30.0.0/16`
DNS Resolution	activado

1. Dentro de la VCN, cree un Internet Gateway con **Create Internet Gateway**, nombre `lab01-igw`.
2. En **Route Tables**, edite la tabla por defecto y agregue una regla:

Campo	Valor
Target Type	Internet Gateway
Destination CIDR Block	`0.0.0.0/0`
Target	`lab01-igw`

1. En **Security Lists**, entre a la lista por defecto y **no la toque**. Esa es la gracia: OCI la crea con el puerto 22 abierto a `0.0.0.0/0`, y así se queda.
2. Cree una segunda security list, `lab01-sl-app`, con dos reglas de entrada:

Campo	Valor
Source CIDR	`0.0.0.0/0` · Protocol TCP · Destination Port `80`
Source CIDR	`0.0.0.0/0` · Protocol TCP · Destination Port `22`

1. Cree la subred con **Create Subnet**.

Campo	Valor
Name	`lab01-subnet-public`
CIDR Block	`10.30.1.0/24`
Subnet Access	Public Subnet
Security Lists	`lab01-sl-app`

1. Cree dos instancias en esa subred, con IP pública asignada, imagen Oracle Linux 9 y shape `VM.Standard.E4.Flex` de 1 OCPU. El detalle está en el capítulo 11; aquí la única diferencia es que **sí** se asigna IP pública.
2. Cree un balanceador público en esa misma subred, con un listener HTTP en el 80 y las dos instancias como backends (capítulo 12). **No le ponga WAF.**

Con eso el laboratorio A está listo para producir cinco hallazgos altos.

Este laboratorio **no se queda encendido**. Tiene dos servidores con IP pública y el puerto 22 abierto a internet. En un tenancy de prueba, desechable y vigilado, durante unas horas, es un ejercicio; encendido una semana es un incidente esperando ocurrir, porque los escáneres encuentran un 22 abierto en minutos, no en días. Bórrelo el mismo día (capítulo 20).

7. El laboratorio B — VCN y gateways

A partir de aquí se construye el laboratorio bien hecho, en el compartimento `lab-02-seguridad`.

Consola de OCI Networking › Virtual Cloud Networks › Create VCN

1. Cree la VCN.

Campo	Valor
Name	`lab02-vcn`

que intenta un servidor comprometido es salir, y controlar la salida es lo que limita el daño.

8. Tablas de ruteo

Cada capa tiene su propia tabla. Es el mecanismo que hace que "la capa de datos no sale a internet" sea un hecho de la plataforma y no una nota en un documento.

Consola de OCI Networking › Virtual Cloud Networks › lab02-vcn › Route Tables › Create Route Table

1. Cree la tabla de la capa pública.

Campo	Valor
Name	`lab02-rt-publica`
Target Type	Internet Gateway
Destination CIDR Block	`0.0.0.0/0`
Target	`lab02-igw`

1. Cree la tabla de la capa de aplicación, con **dos** reglas. Use *+ Another Route Rule* para agregar la segunda.

Campo	Valor
Name	`lab02-rt-app`
Regla 1 · Target Type	NAT Gateway
Regla 1 · Destination CIDR Block	`0.0.0.0/0`
Regla 1 · Target	`lab02-nat`
Regla 2 · Target Type	Service Gateway
Regla 2 · Destination	el bloque de servicios de la región (se elige de una lista, no se escribe)
Regla 2 · Target	`lab02-sgw`

1. Cree la tabla de la capa de datos, con **una sola** regla.

Campo	Valor
Name	`lab02-rt-datos`

Target Type	Service Gateway
Destination	el bloque de servicios de la región
Target	`lab02-sgw`

Fíjese en lo que **no** tiene la tabla de datos: ninguna regla hacia `0.0.0.0/0`. No hay ruta. No es que el tráfico se filtre y se descarte; es que no hay camino.

9. Subredes y la security list por defecto

9.1 Endurecer la security list por defecto — primero

Esto va antes de crear las subredes, porque una subred creada sin especificar security list **hereda la por defecto**, y la por defecto que OCI entrega trae el puerto 22 abierto a `0.0.0.0/0`.

Consola de OCI Networking › Virtual Cloud Networks › lab02-vcn › Security Lists › Default Security List for lab02-vcn

1. Abra la lista por defecto y mire sus reglas de entrada. Verá una regla TCP con origen `0.0.0.0/0` y puerto destino 22. Esa regla es el hallazgo de seguridad más repetido de OCI, y casi nadie la revisa porque "esa lista no está asociada a nada".
2. Borre la regla del puerto 22 con el menú de acciones de la fila.
3. Deje o agregue las dos reglas ICMP. Sin ellas, el descubrimiento de MTU de ruta se rompe y las conexiones fallan de una manera difícil de diagnosticar.

Campo	Valor
Regla 1 · Source CIDR	`0.0.0.0/0` · Protocol ICMP · Type `3` · Code `4`
Regla 2 · Source CIDR	`10.40.0.0/16` · Protocol ICMP · Type `3`

1. En las reglas de salida, deje la regla que permite todo hacia `0.0.0.0/0`. El control de salida real lo hacen las tablas de ruteo y los NSG.
2. Renombre la lista a `lab02-sl-default-endurecida` si la consola lo permite, para que se note en el inventario que alguien la tocó.

TIP · La security list por defecto es el hallazgo más repetido

Aparece en el laboratorio A como un hallazgo alto y aparecería en casi cualquier tenancy auditado por primera vez. Tiene tres propiedades que la vuelven peligrosa: se crea sola con cada VCN, no aparece en ninguna revisión de diseño porque nadie la creó, y se hereda en silencio cada vez que alguien crea una subred sin pensar en listas.

La corrección cuesta treinta segundos y es el mejor ejemplo de "control de esfuerzo bajo" que tiene este laboratorio. En el checklist es el control R-02.

9.2 Las tres subredes

Consola de OCI Networking › Virtual Cloud Networks › lab02-vcn › Subnets › Create Subnet

1. Subred pública, solo para el balanceador.

Campo	Valor
Name	`lab02-subnet-publica-lb`
Subnet Type	Regional
IPv4 CIDR Block	`10.40.1.0/24`
Subnet Access	Public Subnet
Route Table	`lab02-rt-publica`
DNS Label	`publica`
Security Lists	la lista por defecto endurecida

1. Subred privada de aplicación.

Campo	Valor
Name	`lab02-subnet-privada-app`
Subnet Type	Regional
IPv4 CIDR Block	`10.40.2.0/24`
Subnet Access	**Private Subnet**
Route Table	`lab02-rt-app`
DNS Label	`app`
Security Lists	la lista por defecto endurecida

1. Subred privada de datos.

Campo	Valor
Name	`lab02-subnet-privada-datos`

Subnet Type	Regional
IPv4 CIDR Block	`10.40.3.0/24`
Subnet Access	**Private Subnet**
Route Table	`lab02-rt-datos`
DNS Label	`datos`
Security Lists	la lista por defecto endurecida

Marcar una subred como *Private Subnet* hace algo más fuerte que "no asignar IP pública por defecto": **la plataforma impide asignarla**. Cuando alguien cree una instancia ahí dentro dentro de seis meses, la opción de IP pública aparecerá deshabilitada, con la subred como explicación. Es un control que sobrevive al olvido.

10. Network Security Groups

Las security lists quedaron casi vacías a propósito. El filtrado real de este laboratorio vive en tres NSG, uno por función.

```
internet --80/443--> [nsg-lb] --80--> [nsg-app] --3306--> [nsg-datos]
                                            ^
                        bastion --22-------'
                        (solo desde la IP privada del endpoint)
```

10.1 Crear los tres grupos

Consola de OCI Networking › Virtual Cloud Networks › lab02-vcn › Network Security Groups › Create Network Security Group

1. Cree `lab02-nsg-lb`. En la pantalla de reglas del asistente, no agregue ninguna todavía: presione siguiente y cree el grupo vacío.
2. Cree `lab02-nsg-app`, también vacío.
3. Cree `lab02-nsg-datos`, también vacío.

Se crean los tres primero porque las reglas se referencian entre sí: la del NSG de aplicación necesita que el NSG del balanceador ya exista para seleccionarlo.

10.2 Reglas del NSG del balanceador

Consola de OCI Networking › Virtual Cloud Networks › lab02-vcn › Network Security Groups › lab02-nsg-lb › Security Rules › Add Rules

1. Entrada desde internet en el 80.

Campo	Valor
Direction	Ingress
Source Type	CIDR
Source	`0.0.0.0/0`
IP Protocol	TCP
Destination Port Range	`80`
Description	HTTP publico hacia el balanceador

1. Entrada desde internet en el 443, con los mismos campos y puerto destino `443`.
2. Salida hacia la aplicación, **y solo hacia la aplicación**.

Campo	Valor
Direction	Egress
Destination Type	**Network Security Group**
Destination	`lab02-nsg-app`
IP Protocol	TCP
Destination Port Range	`80`
Description	Balanceador hacia la app

El tipo de destino *Network Security Group* es la pieza interesante: no se escribe un CIDR ni una IP, se nombra el grupo. Si mañana se agregan cuatro servidores más a la capa de aplicación, esta regla no cambia.

10.3 Reglas del NSG de la aplicación

Consola de OCI Networking › Virtual Cloud Networks › lab02-vcn › Network Security Groups › lab02-nsg-app › Security Rules › Add Rules

1. Entrada en el 80, solo desde el balanceador.

Campo	Valor
Direction	Ingress

Source Type	**Network Security Group**
Source	`lab02-nsg-lb`
IP Protocol	TCP
Destination Port Range	`80`
Description	Solo el balanceador llega a la app

Note lo que esta regla **no** dice: no dice "desde la subred pública". Un recurso cualquiera colocado en la subred pública no alcanza la aplicación; solo la alcanza lo que esté dentro del NSG del balanceador.

1. La regla del puerto 22 se agrega **después**, en el capítulo 14, porque necesita la IP privada del endpoint del bastión, que todavía no existe. Déjela pendiente.

10.4 Reglas del NSG de datos

Consola de OCI Networking › Virtual Cloud Networks › lab02-vcn › Network Security Groups › lab02-nsg-datos › Security Rules › Add Rules

1. Entrada en el puerto de la base, solo desde la aplicación.

Campo	Valor
Direction	Ingress
Source Type	**Network Security Group**
Source	`lab02-nsg-app`
IP Protocol	TCP
Destination Port Range	`3306`
Description	MySQL solo desde la app

En este laboratorio no hay base de datos: la regla está para que la conversación sobre la capa de datos tenga un objeto al que apuntar.

11. Instancias de aplicación, sin IP pública

Dos instancias, en la subred privada de aplicación, con el plugin de bastión habilitado y una aplicación mínima que responde en el 80.

Consola de OCI Compute › Instances › Create Instance

1. Datos básicos.

Campo	Valor
Name	`lab02-app-1`
Compartment	`lab-02-seguridad`
Availability Domain	el primero que ofrezca la región

1. En **Image and shape**, presione *Change image* y *Change shape*.

Campo	Valor
Image	Oracle Linux 9, la más reciente
Shape series	AMD
Shape	`VM.Standard.E4.Flex`
OCPUs	`1`
Memory (GB)	`8`

1. En **Networking**, seleccione la VCN y la subred privada.

Campo	Valor
Virtual cloud network	`lab02-vcn`
Subnet	`lab02-subnet-privada-app`
Public IPv4 address	**no asignar** — la consola lo deshabilita porque la subred lo prohíbe
Hostname	`app1`

1. En la misma sección, active el uso de NSG y seleccione el grupo de la aplicación.

Campo	Valor
Use network security groups to control traffic	activado
Network security group	`lab02-nsg-app`

1. En **Add SSH keys**, pegue el contenido de su llave pública. Aunque el acceso sea por

bastión, la llave se instala igual: el bastión transporta la sesión, no reemplaza la autenticación del sistema operativo.

2. Abra **Show advanced options** y vaya a la pestaña de **Oracle Cloud Agent**. Verifique que el plugin **Bastion** esté habilitado. Si no lo está, actívelo aquí. Sin este plugin no hay sesión SSH administrada, solo port forwarding.
3. En la pestaña de opciones de gestión, en el campo de datos de inicialización (*user data*), pegue la configuración de `cloud-init` del laboratorio o súbala como archivo. Lo que hace es instalar una aplicación mínima en Python que responde:

Ruta	Respuesta
`GET /`	Nombre del host que atendió, número de OCPU y hora
`GET /health`	`200 ok` — lo consulta el health check del balanceador
`GET /burn?ms=250`	Quema CPU durante los milisegundos indicados

1. Presione **Create**. La instancia queda en aprovisionamiento unos minutos.
2. Repita los pasos 1 a 8 para `lab02-app-2`, con hostname `app2`.

11.1 Dos detalles que rompen el laboratorio si se omiten

El cortafuegos del sistema operativo. Oracle Linux trae `firewalld` activo y con el puerto 80 cerrado. Si no se abre, el health check del balanceador nunca pasa y los backends quedan permanentemente en estado crítico, sin ninguna pista de red que lo explique —los NSG están bien, las rutas están bien—. El `cloud-init` del laboratorio lo abre. Si crea la instancia sin `cloud-init`, tendrá que entrar por el bastión y ejecutar:

```
sudo firewall-cmd --permanent --add-port=80/tcp
sudo firewall-cmd --reload
```

El plugin de bastión tarda. Después de que la instancia aparece como *Running*, el plugin necesita unos minutos más para reportarse; diez minutos desde la creación es una espera razonable antes de la primera sesión administrada. Si intenta antes, la sesión falla con un mensaje sobre el recurso destino que no menciona el plugin.

Verificación:

```
oci compute instance list-vnics --instance-id "<ocid-instancia>" \
  --query 'data[0].{privada:"private-ip",publica:"public-ip"}'
# -> publica debe venir en null
```

12. Balanceador de carga

Consola de OCI Networking › Load Balancers › Load Balancer › Create Load Balancer

1. Detalles del balanceador.

Campo	Valor
Load Balancer Name	`lab02-lb`
Visibility Type	Public
Shape	Flexible
Minimum Bandwidth (Mbps)	`10`
Maximum Bandwidth (Mbps)	`100`

1. Red del balanceador. Es la **única** parte de la arquitectura que vive en la subred pública.

Campo	Valor
Virtual Cloud Network	`lab02-vcn`
Subnet	`lab02-subnet-publica-lb`
Use network security groups to control traffic	activado
Network security group	`lab02-nsg-lb`

1. En la pantalla de backends, elija la política y agregue las dos instancias con *Add Backends*.

Campo	Valor
Load balancing policy	Weighted Round Robin
Backends	`lab02-app-1` y `lab02-app-2`, puerto `80`

1. Configure el health check en la misma pantalla.

Campo	Valor
Protocol	HTTP
Port	`80`

URL Path	`/health`
Status Code	`200`
Interval (ms)	`10000`
Timeout (ms)	`3000`
Number of Retries	`3`

1. Configure el listener.

Campo	Valor
Listener Name	`lab02-listener-http`
Type of Traffic	HTTP
Port	`80`

1. Presione **Submit**. El balanceador tarda varios minutos en quedar activo.
2. Anote la IP pública que aparece en la página de detalle. Se usa en los capítulos de WAF y de validación.

12.1 Sobre el listener sin TLS

Este listener es HTTP porque el laboratorio no debe depender de un certificado. En un ambiente con datos, el listener es 443 con certificado gestionado y el 80 únicamente redirige. Es el control B-03 del checklist, y el script de auditoría lo reporta como hallazgo **medio** en los dos laboratorios, incluido el bien construido. Está bien que lo reporte: un hallazgo en el laboratorio "correcto" prueba que el script no está amañado.

13. WAF sobre el balanceador

Aquí está la demostración central del borde: el mismo ataque contra dos balanceadores, uno responde 200 y el otro 403.

El WAF de OCI se compone de dos objetos: una **política** (las reglas) y un **firewall** (la aplicación de esa política sobre un punto de exigencia, en este caso el balanceador). En la consola el asistente los crea casi de corrido, pero conviene saber que son dos cosas, porque en la lista de recursos aparecen separados.

13.1 Las llaves de capacidad y sus versiones — verificar antes

Consola de OCI Identity & Security › Web Application Firewall › Policies › Create WAF policy

Antes de tocar nada, entienda el punto que hace fallar esta parte del laboratorio más que ningún otro.

IMPORTANTE · Cada capacidad de protección lleva su propia versión

Las dos reglas OWASP de este laboratorio **no están en la misma versión**:

| Llave | Qué detecta | Versión | |---|---|---| | `941100` | XSS, mediante libinjection | **2** | | `942100` | Inyección SQL, mediante libinjection | **1** |

En la consola esto se ve al agregar cada capacidad a la regla de protección: cada fila de la tabla de capacidades trae **su propio selector de versión**, y el valor por defecto no es necesariamente el mismo para todas.

Es un error silencioso. Si pone la versión equivocada en una de las dos, la consola acepta la política —la crea, queda activa, el balanceador queda protegido— y la regla equivocada simplemente **no filtra nada**. No hay error, no hay advertencia, no hay estado degradado. El síntoma aparece mucho después, en la prueba: la columna con WAF devuelve 200 donde debería devolver 403.

Las versiones de la tabla se verificaron en `us-chicago-1`. **En otra región pueden diferir**: la lista de capacidades y sus versiones es propia de cada región. Verifíquelas antes de crear la política.

Si tiene la CLI a mano, la verificación es una línea por llave:

```
for k in 941100 942100; do
  oci waf protection-capability list --compartment-id "<ocid-tenancy>" \
    --key "$k" --all \
    --query 'data.items[].{llave:key,version:version,nombre:"display-name"}' \
    --output table
done
```

Si solo tiene consola, la misma información está en la tabla de selección de capacidades del asistente: cada capacidad muestra las versiones disponibles en la región. Anótelas antes de continuar.

13.2 Crear la política

1. Datos básicos de la política.

Campo	Valor

Name	`lab02-waf-policy`
Compartment	`lab-02-seguridad`

1. En el paso de selección del punto de exigencia, elija el balanceador como destino de la política.

Campo	Valor
Tipo de punto de exigencia	Load balancer
Load balancer	`lab02-lb`

1. Defina las dos **acciones** que la política va a usar. Una permite y la otra devuelve un 403 con cuerpo propio.

Campo	Valor
Acción 1 · Name	`permitir`
Acción 1 · Type	Allow
Acción 2 · Name	`bloquear-403`
Acción 2 · Type	Return HTTP response
Acción 2 · Response code	`403`
Acción 2 · Header	`Content-Type` : `text/plain; charset=utf-8`
Acción 2 · Body type	Static text
Acción 2 · Body	`403 - Solicitud bloqueada por el WAF`

El cuerpo propio no es cosmético: cuando alguien pregunte si lo bloqueó el WAF o la aplicación, la respuesta es el texto que aparece en pantalla.

1. Agregue la regla de **control de acceso**, que se evalúa antes que la protección.

Campo	Valor
Rule name	`bloquear-admin`
Condition language	JMESPATH
Condition	`starts_with(http.request.url.path, '/admin')`

Action	`bloquear-403`
Default action de la sección	`permitir`

1. Agregue la regla de **protección** con las dos capacidades OWASP.

Campo	Valor
Rule name	`owasp-xss-sqli`
Action	`bloquear-403`
Body inspection	desactivado
Capacidad 1	`941100` · versión `2`
Capacidad 2	`942100` · versión `1`

Al agregar cada capacidad, **mire la columna de versión de esa fila** y fíjela en el valor que verificó en 13.1. No asuma que la segunda hereda la de la primera.

1. Revise el resumen y presione **Create**. La política y el firewall quedan creados y asociados al balanceador.
2. Espere. La política tarda unos minutos en propagarse al punto de exigencia. Si prueba inmediatamente y todo devuelve 200, no concluya que las versiones están mal: espere cinco minutos y vuelva a probar.

Verificación:

```
oci waf web-app-firewall get --web-app-firewall-id "<ocid-waf>" \
  --query 'data."lifecycle-state"'
# -> "ACTIVE"
```

13.3 Qué hace y qué no hace el WAF

El WAF no reemplaza el código seguro: compra tiempo cuando aparece una vulnerabilidad nueva y reduce el ruido de los ataques automatizados. Los payloads de la validación son de libro de texto; un atacante con intención es más sutil. Conviene decirlo antes de que lo pregunten.

La regla de control de acceso sobre `/admin` es más robusta que las reglas OWASP y no depende de versiones de capacidad: si algo falla, esa fila sigue funcionando.

14. Bastión

El bastión es el capítulo donde el laboratorio hace su afirmación más incómoda: **el puerto 22 abierto a internet no es una necesidad operativa, es una costumbre.**

14.1 Crear el bastión

Consola de OCI Identity & Security › Bastion › Create Bastion

1. Complete el formulario.

Campo	Valor
Bastion name	`lab02bastion`
Compartment	`lab-02-seguridad`
Target virtual cloud network	`lab02-vcn`
Target subnet	`lab02-subnet-privada-app`
CIDR block allowlist	`0.0.0.0/0` — vea el recuadro antes de escribir esto
Maximum session time-to-live	`10800` segundos (3 horas)

El nombre del bastión admite **solo caracteres alfanuméricos**. No acepta guiones, que es por lo que este recurso rompe la convención de nombres del resto del laboratorio. Si escribe `lab02-bastion`, el formulario lo rechaza con un mensaje sobre el formato del nombre.

1. Presione **Create Bastion** y espere a que quede activo.
2. Anote la **IP privada del endpoint** del bastión, que aparece en la página de detalle. Es un valor dentro de `10.40.2.0/24`, y se necesita en el paso 5.

CUIDADO · La lista de CIDR está en 0.0.0.0/0 a propósito, y hay que decir por qué

En este laboratorio la lista de clientes permitidos del bastión está abierta a todo internet **deliberadamente**, porque es un ambiente desechable que se destruye el mismo día y porque la IP pública desde la que se va a demostrar no se conoce de antemano. No es una recomendación y no debe copiarse.

La práctica correcta en un ambiente con datos es la lista restringida a los CIDR desde donde opera el equipo: la salida de la oficina, el rango de la VPN corporativa, la IP del runner de automatización. Una lista de tres entradas, revisada cuando cambia el proveedor de internet. Si el

equipo es remoto y las IP son dinámicas, la respuesta es la VPN, no `0.0.0.0/0`.

Y ahora lo que hay que entender bien: abrir el bastión no es abrir el puerto 22. Son dos cosas distintas y confundirlas lleva a conclusiones equivocadas en las dos direcciones. Quien llegue a este bastión desde cualquier punto de internet todavía necesita, simultáneamente:

- **credenciales válidas de OCI** en este tenancy, con permiso explícito sobre el recurso de bastión (verbo `manage` o `use` sobre `bastion-family` en el compartimento) para poder siquiera crear una sesión; - **la llave privada** correspondiente a la llave pública que se registró **en esa sesión concreta**, no en el bastión ni en la instancia; - que la **sesión esté viva**: cuando el TTL vence, el camino desaparece.

Un puerto 22 abierto a `0.0.0.0/0` sobre una instancia, en cambio, acepta intentos de conexión de cualquiera, sin identidad, sin registro útil y sin caducidad. La diferencia entre las dos situaciones no es de grado.

La evidencia de esto está medida en este mismo laboratorio: después de abrir la lista de clientes del bastión a `0.0.0.0/0`, **la auditoría volvió a dar 5 hallazgos altos contra 0**, exactamente igual que antes. La lista de CIDR del bastión no es una security list, y el control R-02 siguió reportando que ninguna security list abre SSH, RDP o MySQL a internet.

14.2 Cerrar el círculo: la regla del puerto 22

Ahora que existe el endpoint, se puede escribir la regla que faltaba en el capítulo 10.

Consola de OCI Networking › Virtual Cloud Networks › lab02-vcn › Network Security Groups › lab02-nsg-app › Security Rules › Add Rules

1. Agregue la regla de entrada del 22.

Campo	Valor
Direction	Ingress
Source Type	CIDR
Source	la IP privada del endpoint del bastión, con máscara `/32`
IP Protocol	TCP
Destination Port Range	22
Description	SSH solo desde el endpoint del bastion

Una máscara `/32`: un solo host. El puerto 22 de la aplicación existe y es alcanzable, pero desde

exactamente una dirección de la red interna, que a su vez solo transporta sesiones autenticadas y con vencimiento.

14.3 Abrir una sesión

Consola de OCI Identity & Security › Bastion › lab02bastion › Sessions › Create Session

1. Elija el tipo de sesión y complete.

Campo	Valor
Session type	Managed SSH session
Session name	un nombre corto con la hora, para reconocerla
Compute instance	`lab02-app-1`
Username	`opc`
Add SSH key	pegue el contenido de su llave pública
Maximum session time-to-live	hasta el máximo del bastión

1. Presione **Create session**. Tarda de uno a dos minutos.
2. Cuando la sesión quede activa, use el menú de acciones de la fila para copiar el comando SSH que la consola genera. Ese comando trae un `ProxyCommand` que salta por el bastión.
3. Reemplace en el comando el marcador de la llave privada por la ruta a su llave, y ejecútelo.

TIP · Sesión administrada frente a port forwarding, y cuándo falla cada una

El bastión ofrece dos tipos de sesión y no son intercambiables.

Managed SSH session. Usa el plugin de bastión del agente de cómputo dentro de la instancia. Le da una sesión SSH al sistema operativo, con el usuario que indique. Es la que se quiere. **Falla cuando** el plugin no está habilitado en la instancia, cuando la instancia acaba de arrancar y el plugin todavía no se ha reportado (los primeros minutos), o cuando el Service Gateway no cubre todos los servicios de OCI y el agente no puede hablar con el plano de control.

SSH port forwarding session. No usa el plugin: abre un túnel TCP hacia una IP privada y un puerto. Se conecta a `localhost` en un puerto local y el tráfico sale por el bastión. **No depende del plugin**, así que es el respaldo natural cuando la administrada falla. Es también la única opción para llegar a algo que no es una instancia con agente: una base de datos administrada, por

ejemplo. **Falla cuando** la IP privada destino no es alcanzable desde el endpoint del bastión, es decir, cuando falta la regla de NSG del paso 4.

Las dos fallan por la misma causa cuando la causa es la red de origen: si su IP pública actual no está en la lista de CIDR del bastión, la sesión **se crea correctamente** y la conexión no entra. Es el síntoma más desconcertante del laboratorio, porque la consola muestra la sesión en verde. Compare su IP (`curl -s ifconfig.me`) con la lista antes de buscar en otro lado.

Y las dos tienen TTL. La sesión vence y desaparece; no se renueva sola. Tres horas es el máximo de este laboratorio. Si la sesión expiró, recrearla toma uno o dos minutos: no es algo que convenga hacer en medio de una demostración.

14.4 Qué mirar una vez dentro

Ya conectado a la instancia, tres comandos cuentan la historia completa:

```
hostname                        # app1
ip -4 addr show | grep inet     # solo 10.40.2.x — ninguna IP publica
curl -s ifconfig.me             # sale por el NAT, no por una IP propia
```

La última línea suele sorprender: la instancia **sí** alcanza internet —actualiza paquetes, llama a una API— pero por el NAT Gateway. Internet no la alcanza a ella.

15. Flow logs

Sin registro de flujos, después de un incidente no hay forma de reconstruir quién habló con quién. Es un control barato y suele estar apagado.

Consola de OCI Observability & Management › Logging › Log Groups › Create Log Group

1. Cree el grupo de logs.

Campo	Valor
Compartment	`lab-02-seguridad`
Name	`lab02-logs-seguridad`
Description	Registros de seguridad del laboratorio

Consola de OCI Observability & Management › Logging › Logs › Enable Service Log

1. Habilite el log de servicio sobre la subred de aplicación.

Campo	Valor
Compartment	`lab-02-seguridad`
Service	Virtual Cloud Network Flow Logs
Resource	`lab02-subnet-privada-app`
Log Category	la categoría que cubre todos los registros de flujo
Log Name	`lab02-flowlogs-app`
Log Group	`lab02-logs-seguridad`

1. En las opciones avanzadas, fije la retención.

Campo	Valor
Log Retention Duration	`30` días

1. Presione **Enable Log**. Los primeros registros tardan varios minutos en aparecer.
2. Genere tráfico —basta con abrir la IP del balanceador en el navegador unas veces— y vuelva al log. Use **Explore Log** o la búsqueda del propio log para ver las entradas.

Cada entrada trae origen, destino, puerto, protocolo, si el paquete fue aceptado o rechazado, y el momento. En una investigación real esa es la diferencia entre creer que entraron por ahí y saberlo, con hora y dirección.

Treinta días de retención es una decisión de laboratorio. En un ambiente con datos se decide con el equipo legal y el de seguridad, y los registros suelen enviarse a un SIEM o a Logging Analytics para correlacionarlos con los de aplicación y los de Audit.

16. Alertas ante cambios críticos

Un cambio de regla de red a las dos de la mañana debería despertar a alguien, no descubrirse una semana después. Este capítulo conecta dos servicios: Events, que escucha el plano de control, y Notifications, que entrega el correo.

16.1 El tema y la suscripción

Consola de OCI Developer Services › Application Integration › Notifications › Create Topic

1. Cree el tema.

Campo	Valor
Name	`lab02-alertas-seguridad`
Compartment	`lab-02-seguridad`
Description	Cambios criticos de red e IAM

1. Dentro del tema, use **Create Subscription**.

Campo	Valor
Protocol	Email
Email	el buzón que va a recibir las alertas, por ejemplo `alertas@ejemplo.com`

1. **Vaya al correo y haga clic en el enlace de confirmación.** Hasta que lo haga, la suscripción queda en estado pendiente y **no llega ninguna alerta**. Es el paso que más se olvida de todo el manual, y su síntoma —silencio— es indistinguible de "no ha pasado nada".

16.2 La regla de eventos

Consola de OCI Observability & Management › Events Service › Rules › Create Rule

1. Datos de la regla.

Campo	Valor
Display Name	`lab02-cambios-criticos`
Description	Cambios en security lists, NSGs y politicas IAM
Rule Conditions · Condition	Event Type

1. Agregue los tipos de evento. En la consola se eligen por servicio y luego por tipo; los identificadores correspondientes son:

Servicio	Tipo de evento
Networking	`com.oraclecloud.virtualnetwork.updatesecuritylist`
Networking	`com.oraclecloud.virtualnetwork.updatenetworksecuritygroupsecurityrules`
Networking	`com.oraclecloud.virtualnetwork.addnetworksecuritygroupsecurityrules`
Identity	`com.oraclecloud.identitycontrolplane.createpolicy`
Identity	`com.oraclecloud.identitycontrolplane.updatepolicy`
Identity	`com.oraclecloud.identitycontrolplane.deletepolicy`

Si algún tipo de evento no aparece en la lista de su región, quítelo y deje los de security list e IAM. La demostración funciona igual.

1. En **Actions**, agregue la acción de notificación.

Campo	Valor
Action Type	Notifications
Notifications Compartment	`lab-02-seguridad`
Topic	`lab02-alertas-seguridad`

1. Presione **Create Rule**.

16.3 Probar que funciona

1. Vaya a la VCN, entre a la security list por defecto y agregue cualquier regla inocua —por ejemplo, entrada ICMP desde el CIDR de la VCN—. Guarde.
2. Espere uno o dos minutos. El correo debe llegar con el detalle del evento: quién lo hizo, sobre qué recurso, a qué hora.
3. Quite la regla que agregó.

Ese correo es una alerta **reactiva**: avisa después de que el cambio ocurrió. El capítulo siguiente es la versión preventiva del mismo problema.

17. Security Zone

Cloud Guard detecta. Una Security Zone **impide**. La diferencia se resume así: detectar le avisa a las tres de la mañana; prevenir no lo despierta.

Una Security Zone asocia una **receta** de políticas a un **compartimento**. Desde ese momento, cualquier operación en ese compartimento que viole una política de la receta es rechazada por la plataforma en el momento de crearse. No hay remediación posterior, no hay ventana de exposición: el recurso no llega a existir.

17.1 La receta

Consola de OCI Identity & Security › Security Zones › Recipes › Create Recipe

1. Datos de la receta.

Campo	Valor
Name	`lab02-receta-demo`
Compartment	`lab`

1. En la lista de políticas disponibles, marque **solamente** la que prohíbe que los buckets de Object Storage tengan acceso público. Está en la categoría de restricción de acceso público; el nombre exacto en la consola describe la denegación de buckets públicos `[VALIDAR el nombre exacto en su región]`.
2. Presione **Create**.

Por qué solo una política: la receta máxima que Oracle provee exige, entre otras cosas, llaves de Vault propias en buckets y volúmenes. Con esa receta, **hasta el bucket privado del capítulo 17.3 sería rechazado**, y el mensaje del ejercicio —lo que cumple pasa sin fricción, lo que no cumple no pasa— se pierde en un muro indiferenciado.

17.2 La zona

Consola de OCI Identity & Security › Security Zones › Create Security Zone

1. Complete el formulario.

Campo	Valor

Name	`lab02-zona-demo`
Compartment	`lab-02-zona-segura`
Security Zone Recipe	`lab02-receta-demo`

1. Presione **Create**. La zona queda asociada al compartimento vacío que se creó en el capítulo 4.

Se usa un compartimento vacío a propósito. Asociar una zona a un compartimento que ya tiene recursos no borra lo que viola la política, pero sí bloquea operaciones posteriores sobre esos recursos, y eso produce sorpresas. Se planea **antes** de poblar el compartimento.

17.3 Qué bloquea de verdad

La prueba está automatizada en `scripts/40-prueba-security-zone.sh`, que hace exactamente dos cosas y una limpieza:

```
./40-prueba-security-zone.sh "<ocid-lab-02-zona-segura>"
```

Paso	Operación	Resultado esperado
1	Crear un bucket **privado** en el compartimento de la zona	**Se crea.** Lo que cumple la política pasa sin fricción
2	Crear un bucket **público** en el mismo compartimento	**Se rechaza**, con un error que cita la política de la zona
3	Borrar el bucket privado	Limpieza

El mismo par de operaciones se puede hacer por consola: **Storage › Buckets › Create Bucket** en el compartimento `lab-02-zona-segura`, una vez con acceso `NoPublicAccess` y otra marcando la visibilidad pública. La segunda falla, y el mensaje de error nombra la política de la Security Zone que la rechazó.

Lo importante del paso 1 es que **no falle**. Una barrera que bloquea todo no es una barrera, es una pared, y nadie la deja puesta: lo que convence es que el trabajo legítimo pase sin pedir permiso.

Qué no bloquea: la Security Zone actúa sobre **operaciones del plano de control en ese compartimento**. No inspecciona el contenido de los objetos, no evalúa lo que pasa dentro de

una instancia y no alcanza a otros compartimentos.

18. Validación de extremo a extremo

Tres pruebas. Las tres están automatizadas en `scripts/`, y las tres se pueden hacer a mano si no tiene la CLI.

18.1 La auditoría: el contraste en números

La prueba principal del módulo: **el mismo script, contra los dos laboratorios.**

```
cd talleres/02-seguridad/scripts
./10-auditoria-rapida.sh "<ocid-lab-01-elasticidad>" "<ocid-tenancy>"   # el laboratorio A
./10-auditoria-rapida.sh "<ocid-lab-02-seguridad>" "<ocid-tenancy>"   # el laboratorio B
```

El script es de **solo lectura**: ejecuta únicamente operaciones de listar y consultar —no crea, no modifica, no borra—, así que es seguro entregárselo a un tercero para que lo corra contra su propio tenancy con un usuario de solo lectura.

Resultado medido:

	Laboratorio A (el «antes»)	Laboratorio B (el «después»)
Hallazgos altos	**5**	**0**
Medios	3	2
Controles que cumplen	2	**6**

Los cinco altos del laboratorio A, uno por uno:

Control	Hallazgo
B-02	Balanceador público `lab01-lb` sin WAF
R-01	Instancia del pool con IP pública
R-01	Segunda instancia del pool con IP pública
R-02	Security list por defecto de `lab01-vcn` abre el 22 a `0.0.0.0/0`
R-02	Security list `lab01-sl-app` abre el 22 a `0.0.0.0/0`

Y los dos medios que **sí** aparecen en el laboratorio bien construido:

Control	Hallazgo
B-03	El balanceador tiene un listener sin TLS
R-01	La subred pública permite IP pública — válido, porque ahí solo vive el balanceador

VALIDACIÓN · Cómo saber que el laboratorio quedó bien

La auditoría del laboratorio B debe terminar en **0 hallazgos altos**, con estos seis controles en estado `ok`:

| Control | Texto esperado | |---|---| | B-02 | El balanceador tiene WAF | | D-02 | Sin buckets públicos en el compartimento | | L-02 | 1 flow log activo | | R-01 | Ninguna instancia en ejecución tiene IP pública | | R-02 | Ninguna security list abre SSH, RDP o MySQL a internet | | R-03 | Ningún NSG abre SSH a internet |

Si aparece un alto en R-01, quedó una instancia con IP pública: revise en qué subred la creó. Si aparece en R-02, la security list por defecto no se endureció (capítulo 9.1). Si aparece en B-02, el WAF no quedó asociado al balanceador: revise que el punto de exigencia de la política sea `lab02-lb`.

Las salidas quedan guardadas automáticamente en `evidencias/` con fecha y hora. Correr el script contra los dos laboratorios es además la mejor prueba de que el script sirve: debe encontrar problemas en uno y no en el otro.

18.2 La prueba del WAF: 200 contra 403

```
./20-prueba-waf.sh "<ip-lb-laboratorio-A>" "<ip-lb-laboratorio-B>"
```

Cuatro solicitudes contra cada balanceador. Resultado medido:

```
Solicitud                       Sin WAF     Con WAF
Normal          GET /           200         200
XSS             GET /?q=<script>    200         403
SQL injection   GET /?id=1' OR '1'= 200         403
Ruta admin      GET /admin          200         403
```

La primera fila es la que da sentido a las otras tres: el tráfico legítimo pasa igual por los dos lados. Si la fila normal diera 403, el WAF estaría roto, no protegiendo.

A mano, sin script, la misma prueba son cuatro comandos:

```
LB="<ip-del-balanceador-con-waf>"
curl -s -o /dev/null -w '%{http_code}\n' "http://$LB/"
curl -s -o /dev/null -w '%{http_code}\n' "http://$LB/?q=%3Cscript%3Ealert(1)%3C%2Fscript%3E"
curl -s -o /dev/null -w '%{http_code}\n' "http://$LB/?id=1%27%20OR%20%271%27%3D%271"
curl -s -o /dev/null -w '%{http_code}\n' "http://$LB/admin"
```

Si la segunda línea devuelve 200 contra el balanceador con WAF, el problema casi siempre es la versión de la capacidad `941100`: vuelva al capítulo 13.1.

18.3 El bastión hacia una máquina sin IP pública

```
./30-sesion-bastion.sh 3600
```

El script crea la sesión, intenta primero la administrada, cae a port forwarding si la administrada falla, y deja el comando SSH armado y listo para pegar. Resultado medido: conexión establecida a una instancia **sin IP pública**, en la subred `10.40.2.0/24`.

Dos detalles que el script resuelve y que hay que conocer si se hace a mano:

- **La ruta de la llave privada va entre comillas simples**, en una variable aparte. Si el directorio del usuario tiene un espacio —lo normal en Windows— el comando se parte y falla con un mensaje sobre el archivo de identidad inaccesible. Y no pueden ser comillas dobles: el `ProxyCommand` ya viene envuelto en dobles y unas dobles adentro lo cierran antes de tiempo, con un error desconcertante sobre caracteres inválidos en el nombre de usuario remoto.
- **La primera conexión pide aceptar la huella dos veces**, la del bastión y la del salto interno. El script agrega `StrictHostKeyChecking=accept-new`, que acepta huellas nuevas y sigue avisando si una conocida **cambia**, que es el caso que importa.

18.4 La barrera preventiva

```
./40-prueba-security-zone.sh "<ocid-lab-02-zona-segura>"
```

Esperado: el bucket privado se crea, el público se rechaza citando la política. Detalle en el capítulo 17.3.

19. Qué puede salir mal

Errores reales, documentados durante la construcción y el ensayo de este laboratorio. No son hipótesis.

19.1 WAF

Síntoma	Causa	Arreglo
El WAF devuelve 200 al XSS pero 403 a `/admin`	Versión equivocada en la capacidad `941100`. La política se creó sin error y la regla no filtra	Capítulo 13.1: `941100` va en versión **2**, `942100` en versión **1**. Verificar por región
El WAF devuelve 200 a todo, recién creado	La política todavía no se propagó al punto de exigencia	Esperar cinco minutos y repetir
La creación de la política falla mencionando una capacidad de protección	La llave o la versión no existe en esa región	Listar las capacidades de la región y ajustar
La creación de la política falla mencionando la condición	Sintaxis JMESPATH de la regla de `/admin`	Quitar la regla de control de acceso. La demostración pierde una fila, nada más

19.2 Balanceador

Síntoma	Causa	Arreglo
Los backends quedan en estado crítico, permanentemente	`firewalld` bloquea el 80 en las instancias	Entrar por bastión y abrir el puerto (capítulo 11.1)
Los backends quedan en estado crítico	Falta la regla de salida del NSG del balanceador hacia el NSG de la aplicación	Capítulo 10.2, paso 6
El balanceador responde pero la aplicación nunca	La aplicación no arrancó: `cloud-init` no se aplicó	Entrar por bastión y revisar el servicio

19.3 Bastión

Síntoma	Causa	Arreglo
La sesión aparece activa y la conexión no entra	La IP pública de origen no está en la lista de CIDR del bastión	Comparar `curl -s ifconfig.me` con la lista y actualizarla

La sesión administrada falla al crearse	El plugin de bastión aún no se ha reportado	Esperar diez minutos desde el arranque de la instancia, o usar port forwarding
La sesión administrada falla siempre	El Service Gateway cubre solo Object Storage: el agente no alcanza el plano de control	Capítulo 7, paso 4: elegir todos los servicios de la región
El puerto 22 no responde a través del túnel	Falta la regla de NSG desde la IP del endpoint	Capítulo 14.2
La creación del bastión falla por el nombre	El nombre admite solo alfanuméricos	Usar `lab02bastion`, sin guiones
«Identity file not accessible» al pegar el comando	La ruta de la llave tiene un espacio y va sin comillas	Comillas **simples**, en variable aparte (capítulo 18.3)
«remote username contains invalid characters»	Se usaron comillas dobles dentro del `ProxyCommand`, que ya va entre dobles	Comillas simples
La conexión funcionaba y dejó de funcionar	El TTL de la sesión venció	Crear una sesión nueva: uno o dos minutos

19.4 Cloud Guard, alertas y zona

Síntoma	Causa	Arreglo
Cloud Guard no se habilita por CLI, con un error de autorización	Faltan las políticas de servicio, que la consola crea sola	Habilitarlo por consola (capítulo 5)
Cloud Guard habilitado y la pantalla de problemas vacía	Todavía no ha hecho un barrido, o los recursos son muy recientes	Dejar pasar horas. Habilitarlo antes de construir
No llega ningún correo de alerta	La suscripción quedó pendiente: falta el clic en el correo de confirmación	Capítulo 16.1, paso 3
La regla de eventos no se puede crear	Algún tipo de evento no se reconoce en la región	Quitar los tipos de NSG y dejar security lists e IAM
La Security Zone rechaza también el bucket privado	La receta es más estricta de lo esperado: exige llaves de Vault	Usar una receta propia con una sola política (capítulo 17.1)

El bucket público **sí** se crea	La receta no incluye la política de denegación de buckets públicos	Revisar la receta y volver a asociarla

19.5 Construcción en general

Síntoma	Causa	Arreglo
El compartimento recién creado no aparece en un formulario	Todavía se está propagando	Esperar dos minutos y recargar
No se puede crear el balanceador o la segunda instancia	Límite de la cuenta de prueba	Construir un laboratorio a la vez, o bajar a una instancia
La instancia no ofrece asignar IP pública	La subred es privada. **Funciona como debe**	No es un error
Falta `jq` al correr el script de auditoría	Dependencia no instalada	`winget install jqlang.jq` en Windows, o el gestor de paquetes del sistema

20. Limpieza

El orden importa: hay recursos que bloquean el borrado de otros.

20.1 Orden de borrado

1. **Sesiones activas del bastión** — Identity & Security › Bastion › lab02bastion › Sessions. Una sesión viva bloquea el borrado del bastión durante varios minutos, con un mensaje sobre el recurso en uso.
2. **El firewall de aplicación y después su política** — Identity & Security › Web Application Firewall. Al revés no se deja.
3. **El balanceador** — Networking › Load Balancers › lab02-lb › Delete. Esto libera los backends.
4. **Las instancias** — Compute › Instances. Marque la opción de borrar también el volumen de arranque; si no la marca, los volúmenes quedan.
5. **El bastión.**
6. **Los logs y el grupo de logs.** El log de flujos primero, el grupo después.
7. **La regla de eventos, la suscripción y el tema de notificaciones.**

8. **La red, en este orden:** subredes, NSG, tablas de ruteo, gateways y por último la VCN. Si un objeto se niega a borrarse, algo lo está referenciando; la consola dice qué.
9. **El laboratorio A**, completo, con la misma secuencia.
10. **Los compartimentos**, si ya no los va a usar. Un compartimento vacío no cuesta nada, pero deja ruido en el inventario.

Si tiene el módulo de Terraform, todo lo anterior es `scripts/99-destroy.sh`, que además cierra las sesiones del bastión antes de destruir.

20.2 Qué se queda, y por qué está bien

Recurso	Se queda	Costo
Cloud Guard	Sí, encendido	Sin costo adicional
Security Zone y su receta	Sí	Sin costo adicional
Compartimento de la zona	Sí, vacío	Sin costo
Presupuesto y alertas de costo	Sí	Sin costo

CUIDADO · Lo que sigue cobrando si la limpieza queda a medias

El balanceador flexible y el WAF consumen crédito **aunque no reciban una sola solicitud**: se cobran por estar aprovisionados, no por tráfico. Son, con diferencia, lo más caro de este laboratorio.

Las instancias detenidas dejan de cobrar cómputo, pero **sus volúmenes de arranque siguen cobrando almacenamiento**. Si al borrar una instancia no marcó la opción de borrar el volumen, el volumen sigue ahí y sigue cobrando: revise **Storage › Block Volumes** en el compartimento después de la limpieza.

Las IP públicas reservadas, si llegó a crear alguna, cobran mientras no estén asignadas a nada. Revise **Networking › IP Management**.

El orden de magnitud de este laboratorio es de unidades de dólar: el ensayo completo de los cinco laboratorios del conjunto midió **0,13 USD** de cómputo al momento de cerrar, con un presupuesto que proyectaba **2,10 USD** contra un límite de 150. Esa cifra es baja porque el teardown fue diario. Un balanceador con WAF olvidado durante un mes es otra conversación.

20.3 Verificar que quedó limpio

Consola de OCI Governance & Administration › Tenancy Explorer

Filtre por el compartimento `lab` y sus hijos: debería quedar vacío salvo los compartimentos mismos. Para el costo, **Billing & Cost Management › Cost Analysis**, filtrando por la etiqueta `Efimero = si`; los datos de consumo tardan horas en consolidar, así que la cifra del mismo día siempre será menor que la real.

21. Documentación oficial

Enlaces a las páginas generales de cada servicio. Las rutas profundas de la documentación cambian entre versiones; desde estas páginas la navegación lateral lleva al procedimiento exacto.

Tema	Enlace
Redes virtuales (VCN, subredes, gateways)	https://docs.oracle.com/en-us/iaas/Content/Network/Concepts/overview.htm
Network Security Groups	https://docs.oracle.com/en-us/iaas/Content/Network/Concepts/networksecuritygroups.htm
Cómputo	https://docs.oracle.com/en-us/iaas/Content/Compute/Concepts/computeoverview.htm
Load Balancer	https://docs.oracle.com/en-us/iaas/Content/Balance/Concepts/balanceoverview.htm
Web Application Firewall	https://docs.oracle.com/en-us/iaas/Content/WAF/Concepts/overview.htm
Bastion	https://docs.oracle.com/en-us/iaas/Content/Bastion/Concepts/bastionoverview.htm
Logging y flow logs	https://docs.oracle.com/en-us/iaas/Content/Logging/Concepts/loggingoverview.htm
Events Service	https://docs.oracle.com/en-us/iaas/Content/Events/Concepts/eventsoverview.htm
Notifications	https://docs.oracle.com/en-us/iaas/Content/Notification/Concepts/notificationoverview.htm

Audit	https://docs.oracle.com/en-us/iaas/Content/Audit/Concepts/auditoverview.htm
Cloud Guard	https://docs.oracle.com/en-us/iaas/cloud-guard/home.htm
Security Zones	https://docs.oracle.com/en-us/iaas/security-zone/home.htm
Identidad y políticas	https://docs.oracle.com/en-us/iaas/Content/Identity/Concepts/overview.htm
Límites de servicio	https://docs.oracle.com/en-us/iaas/Content/General/Concepts/servicelimits.htm

Referencia externa para la conversación de cumplimiento: el **CIS OCI Foundations Benchmark**, que da una línea base reconocida para medir avance y responder auditorías. El script de auditoría de este laboratorio es una muestra deliberadamente pequeña: para postura continua se usa Cloud Guard, y para cumplimiento, el benchmark.