Avance Final del proyecto (Red Team vs Blue Team en Azure)

Integrantes:

Betzabeth Araya Abarca

Jose Arias Rodríguez

Carlos Garreta Quesada

Luis Ugalde Álvarez

Jose Ugalde Moreno

Fiorella Ureña Jaubert

Curso: Programación Avanzada

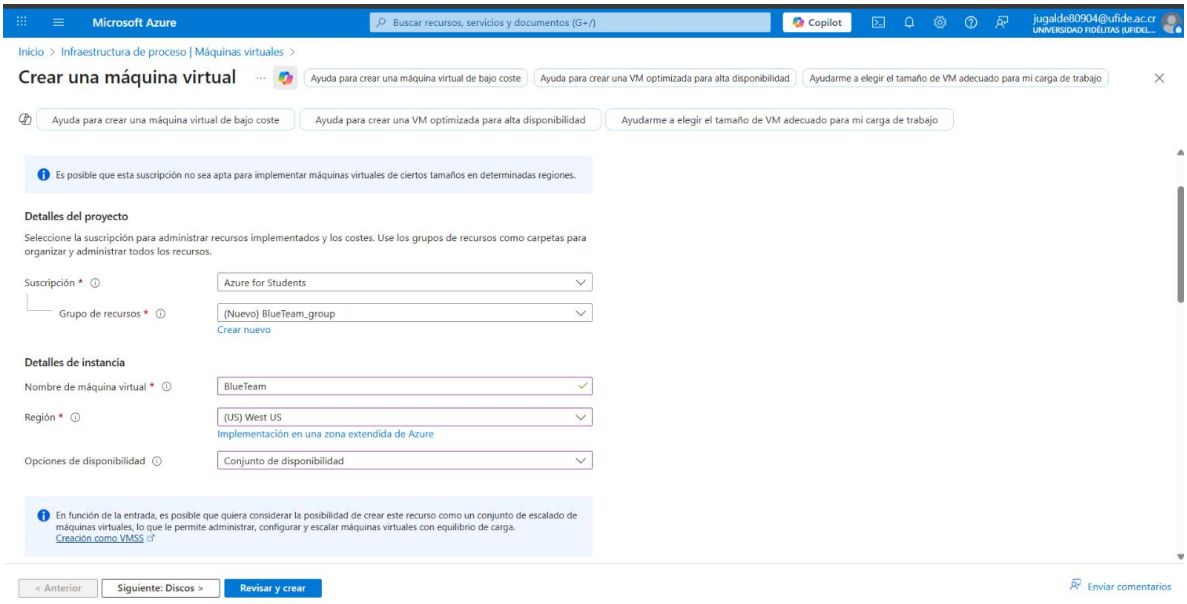
Profesor. Andrés Felipe Vargas Rivera

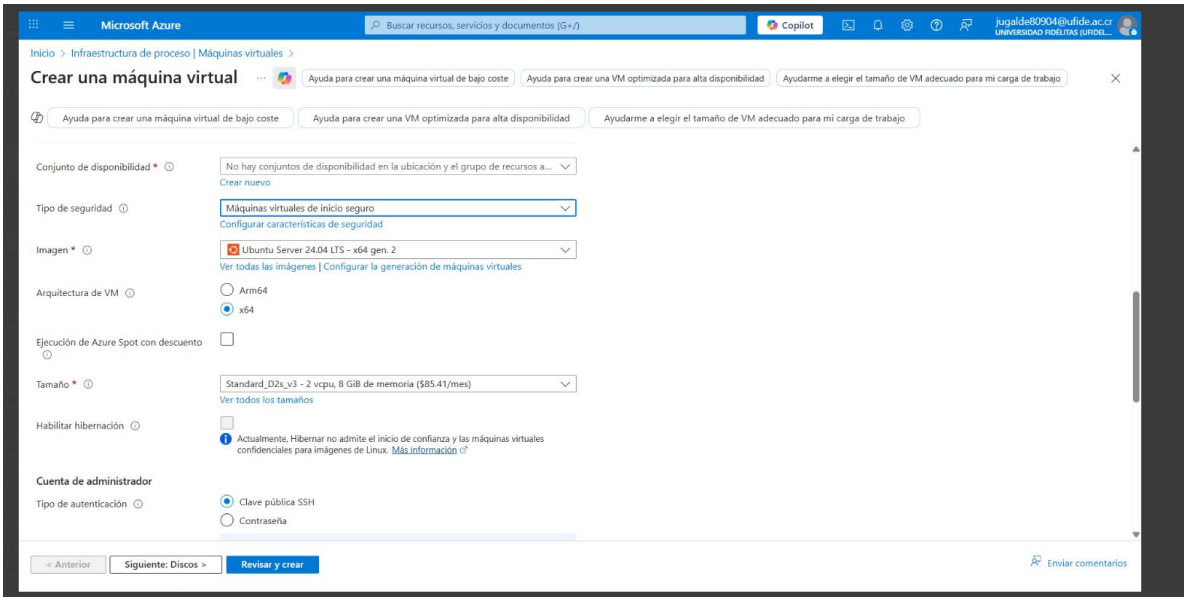
III Cuatrimestre 2025

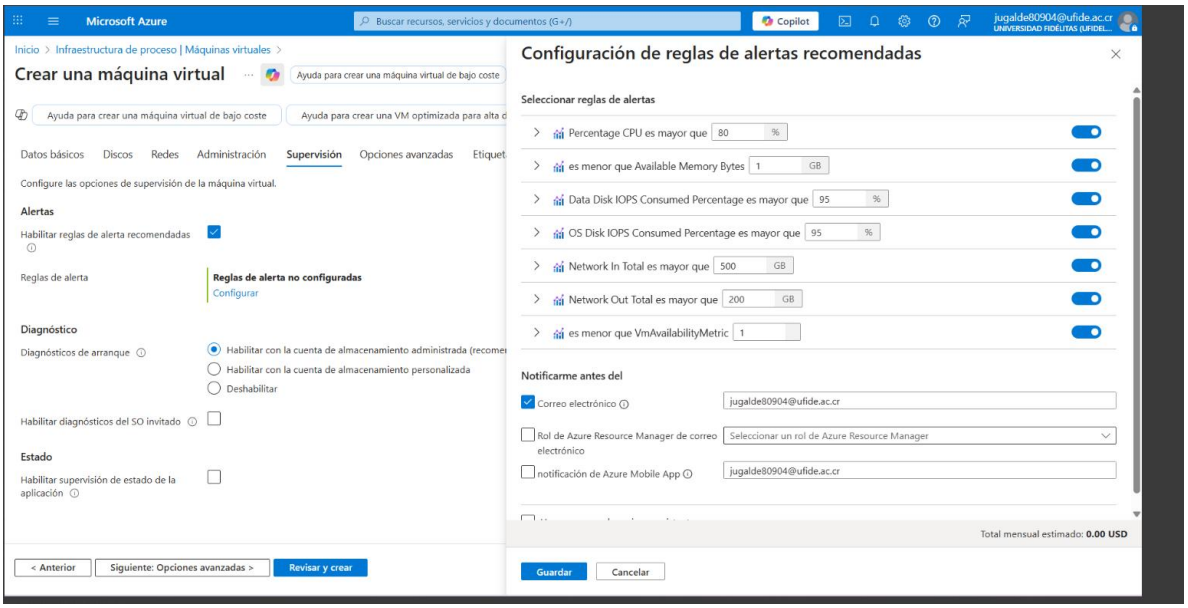
1.Roles del Equipo:

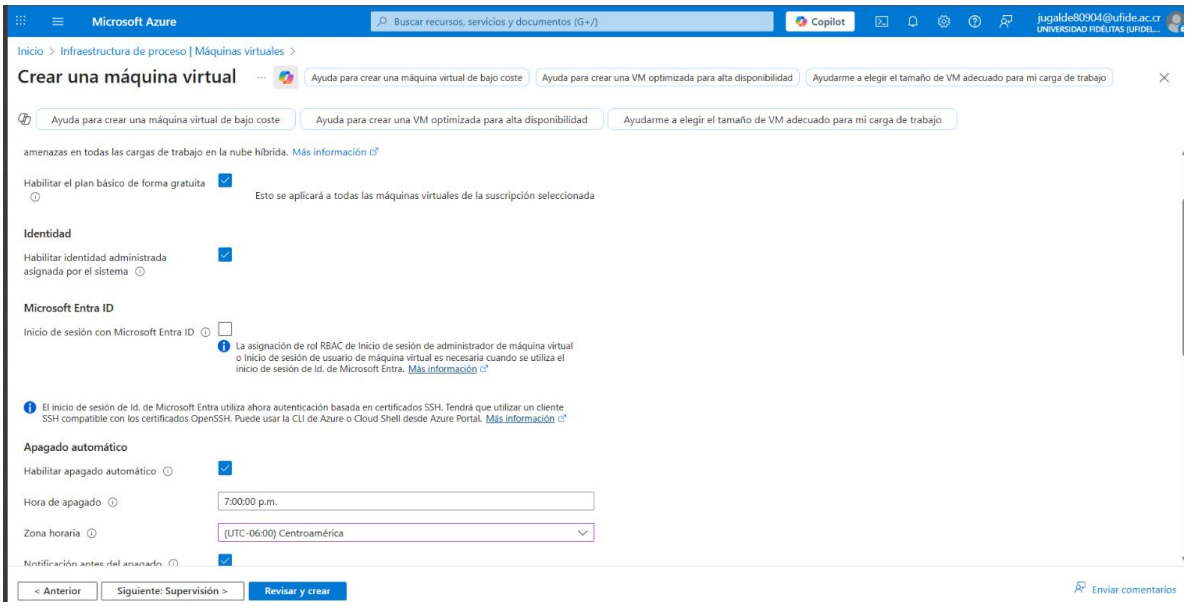
-Blue Team: Su rol es la de protector de ciberataques necesita conocer las formas y métodos de ataque para poder defenderse de los mismos. También hay medidas de que reducen la probabilidad de de ataques.

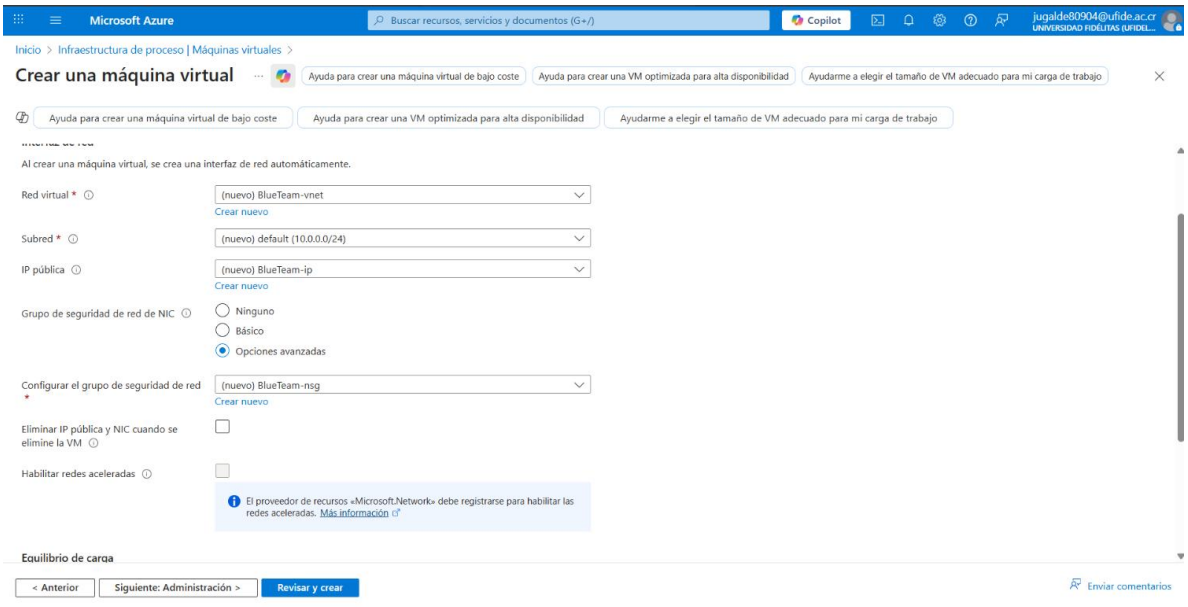
Crear y configurar la VM en Azure

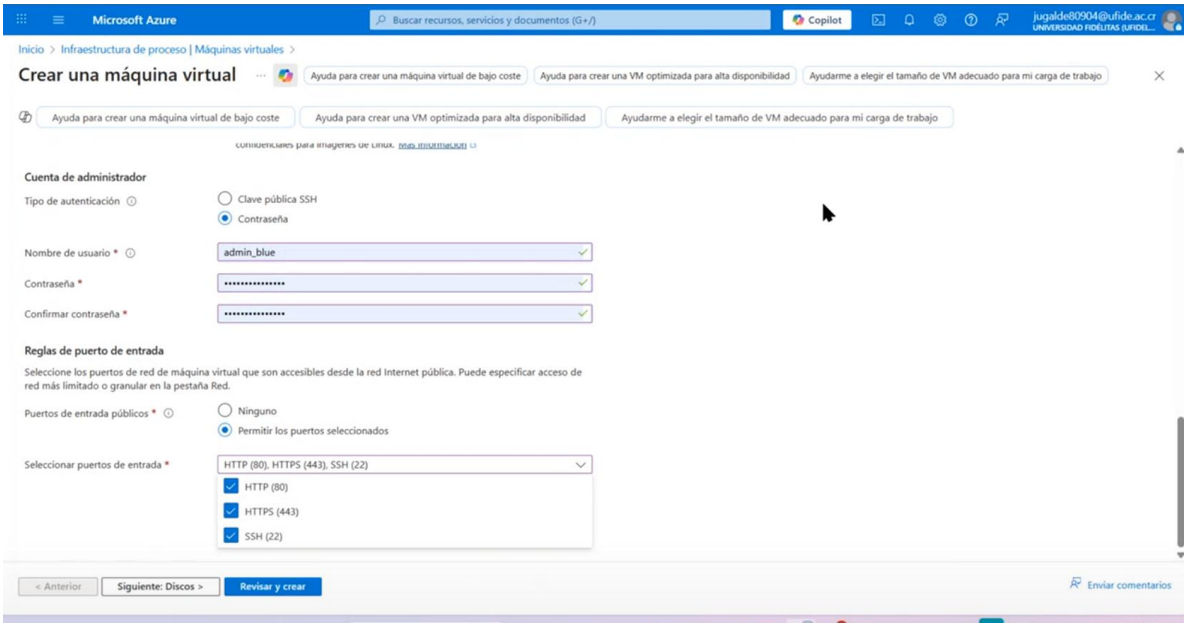


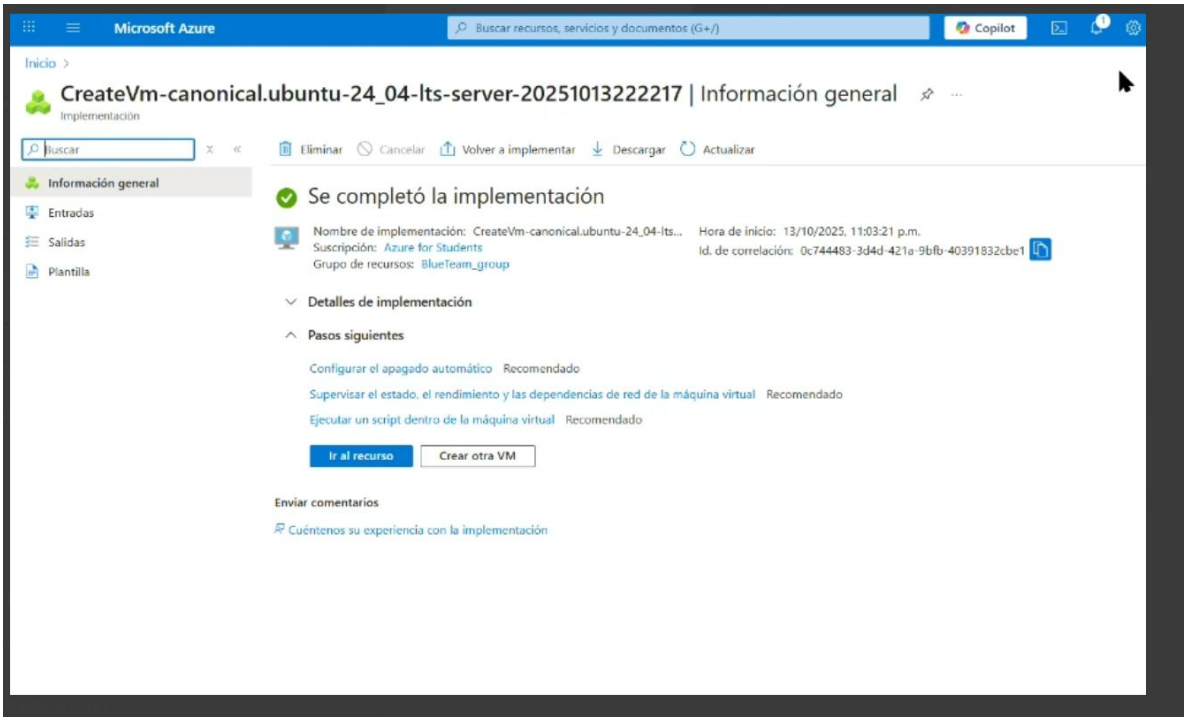



Microsoft Azure
Buscar recursos, servicios y documentos (G+/)
Copilot
jugalde80904@ufide.ac.cr
UNIVERSIDAD FIDÉLITAS (UFIDEL...
Inicio > Infraestructura de proceso | Máquinas virtuales >
Crear una máquina virtual
Ayuda para crear una máquina virtual de bajo coste
Ayuda para crear una VM optimizada para alta d
Datos básicos
Discos
Redes
Administración
Supervisión
Opciones avanzadas
Etiquet
Configure las opciones de supervisión de la máquina virtual.
Alertas
Habilitar reglas de alerta recomendadas
Reglas de alerta
Reglas de alerta no configuradas
Configurar
Diagnóstico
Diagnósticos de arranque
Habilitar con la cuenta de almacenamiento administrada (recome
Habilitar con la cuenta de almacenamiento personalizada
Deshabilitar
Habilitar diagnósticos del SO invitado
Estado
Habilitar supervisión de estado de la aplicación
< Anterior
Siguiente: Opciones avanzadas >
Revisar y crear
Configuración de reglas de alertas recomendadas
Seleccionar reglas de alertas
Percentage CPU es mayor que 80 %
es menor que Available Memory Bytes 1 GB
Data Disk IOPS Consumed Percentage es mayor que 95 %
OS Disk IOPS Consumed Percentage es mayor que 95 %
Network In Total es mayor que 500 GB
Network Out Total es mayor que 200 GB
es menor que VmAvailabilityMetric 1
Notificarme antes del
Correo electrónico
jugalde80904@ufide.ac.cr
Rol de Azure Resource Manager de correo electrónico
Seleccionar un rol de Azure Resource Manager
notificación de Azure Mobile App
jugalde80904@ufide.ac.cr
Total mensual estimado: 0.00 USD
Guardar
Cancelar


Microsoft Azure
Buscar recursos, servicios y documentos (G+/)
Copilot
jugalde80904@ufide.ac.cr
UNIVERSIDAD FIDÉLITAS (UFIDEL...
Inicio > Infraestructura de proceso | Máquinas virtuales >
Crear una máquina virtual
Ayuda para crear una máquina virtual de bajo coste
Ayuda para crear una VM optimizada para alta disponibilidad
Ayudarme a elegir el tamaño de VM adecuado para mi carga de trabajo
amenazas en todas las cargas de trabajo en la nube híbrida. Más información
Habilitar el plan básico de forma gratuita
Esto se aplicará a todas las máquinas virtuales de la suscripción seleccionada
Identidad
Habilitar identidad administrada asignada por el sistema
Microsoft Entra ID
Inicio de sesión con Microsoft Entra ID
La asignación de rol RBAC de Inicio de sesión de administrador de máquina virtual o Inicio de sesión de usuario de máquina virtual es necesaria cuando se utiliza el inicio de sesión de Id. de Microsoft Entra. Más información
El inicio de sesión de Id. de Microsoft Entra utiliza ahora autenticación basada en certificados SSH. Tendrá que utilizar un cliente SSH compatible con los certificados OpenSSH. Puede usar la CLI de Azure o Cloud Shell desde Azure Portal. Más información
Apagado automático
Habilitar apagado automático
Hora de apagado
7:00:00 p.m.
Zona horaria
(UTC-06:00) Centroamérica
Notificación antes del apagado
< Anterior
Siguiente: Supervisión >
Revisar y crear
Enviar comentarios


Microsoft Azure
Buscar recursos, servicios y documentos (G+/)
Copilot
jugalde80904@ufide.ac.cr
UNIVERSIDAD FIDÉLITAS (UFIDEL...
Inicio > Infraestructura de proceso | Máquinas virtuales >
Crear una máquina virtual
Ayuda para crear una máquina virtual de bajo coste
Ayuda para crear una VM optimizada para alta disponibilidad
Ayudarme a elegir el tamaño de VM adecuado para mi carga de trabajo
Al crear una máquina virtual, se crea una interfaz de red automáticamente.
Red virtual *
(nuevo) BlueTeam-vnet
Crear nuevo
Subred *
(nuevo) default (10.0.0.0/24)
IP pública
(nuevo) BlueTeam-ip
Crear nuevo
Grupo de seguridad de red de NIC
Ninguno
Básico
Opciones avanzadas
Configurar el grupo de seguridad de red *
(nuevo) BlueTeam-nsg
Crear nuevo
Eliminar IP pública y NIC cuando se elimine la VM
Habilitar redes aceleradas
El proveedor de recursos «Microsoft.Network» debe registrarse para habilitar las redes aceleradas. Más información
Equilibrio de carga
< Anterior
Siguiente: Administración >
Revisar y crear
Enviar comentarios





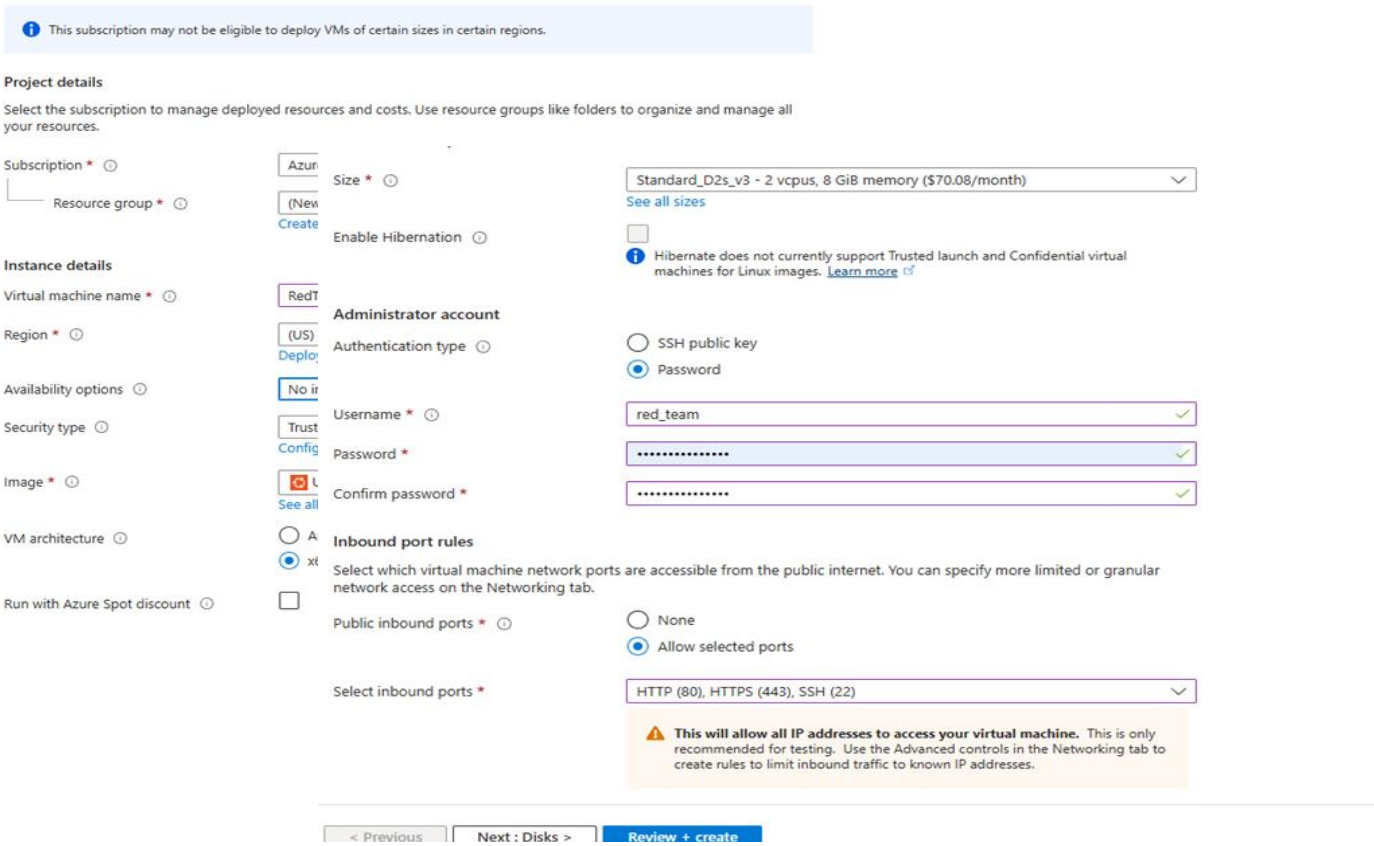
Pasos a seguir:

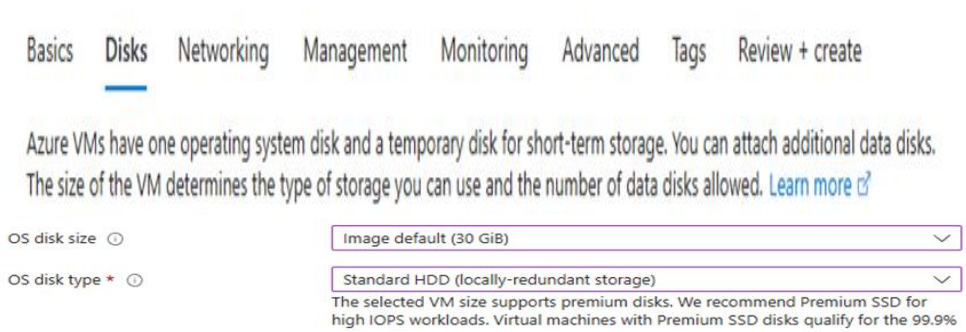
- Desde el navegador con Azure Bastion (sin abrir puertos)
- En el portal de la VM → botón Conectar → pestaña Bastion.
- Si no está creado, haz Crear Bastion (sigue el asistente).
- Rellena usuario y contraseña/clave → Conectar.

  Resultado: consola SSH en el navegador.

-Red Team: Su rol es el de agresor virtual identifica y utiliza las vulnerabilidades del sistema o de los usuarios del sistema. Para modificar leer o ver datos confidenciales.

Creación y Configuración de la VM en Azure



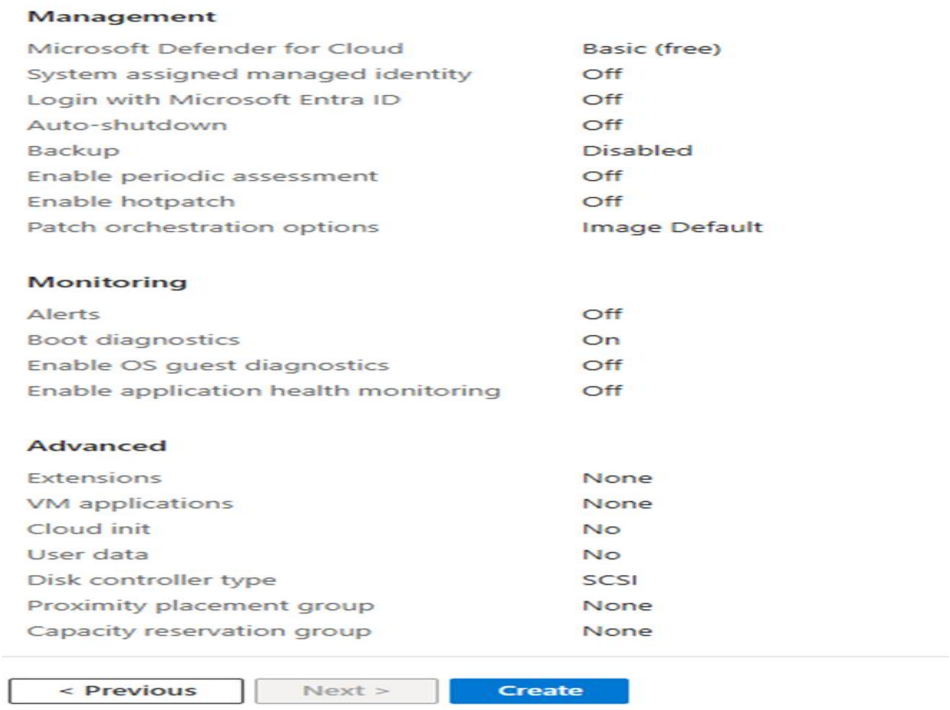


Basics

Subscription	Azure for Students
Resource group	(new) RedTeam_group
Virtual machine name	RedTeam
Region	West US 3
Availability options	No infrastructure redundancy required
Zone options	Self-selected zone
Security type	Trusted launch virtual machines
Enable secure boot	Yes
Enable vTPM	Yes
Integrity monitoring	No
Image	Ubuntu Server 24.04 LTS - Gen2
VM architecture	x64
Size	Standard D2s v3 (2 vcpus, 8 GiB memor
Enable Hibernation	No
Authentication type	Password
Username	red_team
Public inbound ports	SSH, HTTPS, HTTP
Azure Spot	No

Disks

OS disk size	Image default
OS disk type	Standard HDD LRS
Use managed disks	Yes
Delete OS disk with VM	Enabled
Ephemeral OS disk	No



Pasos a seguir:

- Desde el navegador con Azure Bastion (sin abrir puertos)

- En el portal de la VM → botón Conectar → pestaña Bastion.
- Si no está creado, haz Crear Bastion (sigue el asistente).
- Rellena usuario y contraseña/clave → Conectar.

  Resultado: consola SSH en el navegador.

2. IP de la Máquina Virtual Objetivo (BlueTeam):

Máquina Virtual Blue Team:

IP: 172.184.103.17

3. Puertos Permitidos (NSG):

HTTP: 80

HTTPS: 443

SSH: 22

4. Buenas Prácticas de Seguridad:

-Mantener actualizado el sistema con `sudo apt update && sudo apt upgrade -y`.

- Apagar la VM cuando no esté en uso.

- Evitar exponer servicios innecesarios.

- Utilizar contraseñas y claves seguras.

- Supervisar accesos SSH desde direcciones IP conocidas

red_team/scann.py

Corre un script importando las librerías nmap (para escaneo de red) y os (para detectar la carpeta actual) y se realiza un scan con diferentes flags.

Se especifican los diferentes tipos de escaneo que vamos a realizar en la variable “options”:

- -sS: escaneo TCP SYN
- -sV: detecta la versión del servicio
- -Pn: tratar a todos los hosts como online (se salta el ping)
- -p 1-1000: puertos a escanear

README.md – Red Team: Ataque y Evaluación de Seguridad

Propósito

Este módulo documenta las herramientas, técnicas y procedimientos utilizados por el Red Team para identificar vulnerabilidades, ejecutar ataques controlados y evaluar la postura de seguridad de la máquina virtual del Blue Team en Azure.

Roles y Responsabilidades

- Simular ataques reales de ciberseguridad.
- Identificar debilidades en la configuración de la VM.
- Documentar hallazgos y proponer mejoras.
- Utilizar scripts en Python para automatizar escaneos y ataques.

Requisitos Técnicos

- Python 3.x
- Módulos: nmap, scapy, paramiko, os, subprocess, logging
- Acceso a una VM en la misma red que el Blue Team
- Azure CLI (opcional para gestión de recursos)

Scripts Ofensivos

Script	**Descripción**
scanner.py	Escaneo de puertos y servicios usando Nmap desde Python.
packet_attack.py	Ataques de red como sniffing y ARP Spoofing con Scapy.
ssh_brute.py	Ataque de diccionario al servicio SSH usando Paramiko.
report.md	Documentación detallada del ataque, hallazgos y recomendaciones.

Instrucciones de Ejecución

1. scanner.py

sudo python3 scanner.py

- Realiza escaneo básico y avanzado.
- Genera reporte de puertos abiertos y servicios activos.

2. packet_attack.py

sudo python3 packet_attack.py

- Ejecuta sniffing de paquetes TCP.
- Simula ataques ARP/DNS spoofing.

3. ssh_brute.py

python3 ssh_brute.py

- Requiere diccionario de contraseñas.
- Intenta acceso por fuerza bruta al puerto 22.

Evaluación del Éxito

- Acceso no autorizado a la VM del Blue Team.
- Identificación de puertos inseguros o servicios expuestos.
- Captura de tráfico sensible.
- Documentación clara de vulnerabilidades y recomendaciones.

Estructura del Directorio

red_team/

```
│
├── scanner.py
├── packet_attack.py
├── ssh_brute.py
└── report.md
```

Buenas Prácticas

- No ejecutar ataques fuera del entorno controlado.
- Documentar cada paso y resultado.
- Validar que las herramientas estén correctamente instaladas.
- Apagar la VM cuando no esté en uso para conservar créditos de Azure.

README – Blue Team: Defensa y Monitoreo en Azure

Propósito

Este módulo documenta las herramientas, técnicas y procedimientos utilizados por el Blue Team para proteger, monitorear y mantener la seguridad de la máquina virtual en Azure.

Roles y Responsabilidades

- Implementar medidas de defensa y mitigación.
- Monitorear eventos y alertas de seguridad.
- Responder a incidentes y realizar análisis forense.
- Utilizar scripts en Python para automatización y auditoría.

Requisitos Técnicos

- Python 3.x
- Módulos: psutil, os, subprocess, logging, paramiko
- Acceso a la VM en Azure
- Azure CLI para gestión y monitoreo

Scripts Defensivos

Script	**Descripción**
monitor.py	Monitoreo de procesos, uso de CPU y memoria en tiempo real.
log_audit.py	Auditoría y análisis de logs del sistema para detectar anomalías.
alert.py	Envío de alertas por correo o mensajes ante eventos sospechosos.

response.py	Automatización de respuestas a incidentes comunes.

Instrucciones de Ejecución

1. monitor.py

python3 monitor.py

- Monitorea recursos del sistema y genera reportes.

2. log_audit.py

python3 log_audit.py

- Analiza logs para detectar patrones inusuales.

3. alert.py

python3 alert.py

- Configura y envía alertas en tiempo real.

4. response.py

python3 response.py

- Ejecuta acciones automáticas para mitigar amenazas.

Evaluación del Éxito

- Detección temprana de ataques o anomalías.
- Respuesta rápida y efectiva a incidentes.
- Mantenimiento de la integridad y disponibilidad de la VM.
- Documentación clara de eventos y acciones tomadas.

Estructura del Directorio

blue_team/

├── monitor.py

├── log_audit.py

├── alert.py

└── response.py

Buenas Prácticas

- Mantener actualizadas las herramientas y scripts.
- Revisar periódicamente los logs y alertas.
- Documentar todas las acciones y hallazgos.
- No ejecutar scripts en producción sin pruebas previas.
- Asegurar la comunicación segura entre componentes.Azure.

Evidencias de ejecución:

BlueTeam (defensa)

sudo ./firewall_basic.sh

```
BlueTeam@BlueTeam:~$ sudo ./firewall_basic.sh
Reseteando UFW a estado por defecto (si está instalado)...
Backing up 'user.rules' to '/etc/ufw/user.rules.20251217_032402'
Backing up 'before.rules' to '/etc/ufw/before.rules.20251217_032402'
Backing up 'after.rules' to '/etc/ufw/after.rules.20251217_032402'
Backing up 'user6.rules' to '/etc/ufw/user6.rules.20251217_032402'
Backing up 'before6.rules' to '/etc/ufw/before6.rules.20251217_032402'
Backing up 'after6.rules' to '/etc/ufw/after6.rules.20251217_032402'

Estableciendo políticas por defecto: deny incoming, allow outgoing
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
Habilitando logging (low) y habilitando UFW
```

sudo python3 ./os_audit.py

```
BlueTeam@BlueTeam:~$ sudo python3 os_audit.py
Auditoría básica del sistema


--- Usuarios del sistema ---
Usuario: BlueTeam, Shell: /bin/bash

--- Puertos abiertos ---
Netid State  Recv-Q Send-Q Local Address:Port Peer Address:PortProcess
udp   UNCONN 0      0         127.0.0.54:53       0.0.0.0:*
udp   UNCONN 0      0      127.0.0.53%lo:53       0.0.0.0:*
udp   UNCONN 0      0      10.0.0.4%eth0:68       0.0.0.0:*
udp   UNCONN 0      0          127.0.0.1:323      0.0.0.0:*
udp   UNCONN 0      0              [::1]:323         [::]:*
```

sudo python3 ./alert_logger.py

```
BlueTeam@BlueTeam:~$ sudo python3 alert_logger.py
=== Alert Logger del Blue Team ===
Esperando alertas desde otros módulos...

Realizando pruebas locales del logger:

/home/BlueTeam/alert_logger.py:34: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes
  "timestamp": datetime.utcnow().isoformat(),
[INFO] selftest: Inicio del módulo de alertas.
[WARNING] selftest: Actividad sospechosa detectada.
[AUTO] Advertencia registrada. No se requiere acción inmediata.
[CRITICAL] selftest: Posible intrusión detectada.
[AUTO] ALERTA CRÍTICA. Requiere intervención manual.
[AUTO] Detalle: Posible intrusión detectada.

[OK] alert_logger.py está funcionando correctamente.
```

sudo python3 ./sniffer_deffence.py

```
Solo se mostraran alertas CRITICAS (port scans)
Las alertas de puertos sensibles se guardan en log
Ejecutando... (Ctrl+C para detener)
================================================

/home/BlueTeam/sniffer_defense.py:68: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use tim
  now = datetime.utcnow()
[WARNING] sniffer_defense: Trafico a puerto sensible 22 (TCP) desde 177.93.10.96 hacia 10.0.0.4
[WARNING] sniffer_defense: Trafico a puerto sensible 80 (TCP) desde 10.0.0.4 hacia 168.63.129.16
[WARNING] sniffer_defense: Trafico a puerto sensible 22 (TCP) desde 52.225.31.164 hacia 10.0.0.4
[CRITICAL] sniffer_defense: POSIBLE PORT SCAN desde 168.63.129.16. Contacto 20 puertos en 60 segundos.
[AUTO] ALERTA CRÍTICA. Requiere intervención manual.
[AUTO] Detalle: POSIBLE PORT SCAN desde 168.63.129.16. Contacto 20 puertos en 60 segundos.
[WARNING] sniffer_defense: Trafico a puerto sensible 80 (TCP) desde 52.225.31.164 hacia 10.0.0.4
[WARNING] sniffer_defense: Trafico a puerto sensible 443 (TCP) desde 52.225.31.164 hacia 10.0.0.4
```

RedTeam (ataque)

sudo python3 ./scaner.py

```
RedTeam@RedTeam:~$ sudo python3 scanner.py
Host: 20.228.97.25
State: up
Protocol: tcp
Port: 22 State: open
Port: 80 State: closed
Port: 443 State: closed
RedTeam@RedTeam:~$ cat report.txt
Host: 20.228.97.25
State: up
Protocol: tcp
Port: 22 State: open
Port: 80 State: closed
Port: 443 State: closed
```

sudo python3 ./packet_attack.py

```
RedTeam@RedTeam:~$ sudo python3 packet_attack.py
=== Envío seguro de paquetes Scapy (Azure Lab) ===

[~] Enviando ping (ICMP Echo)...
[X] Sin respuesta al ping

[~] Enviando packet TCP SYN seguro...
[X] No hubo respuesta TCP

[~] Enviando paquete UDP de laboratorio...
[OK] Paquete UDP enviado (no disruptivo)

[OK] Simulación completada. Revisa packet_lab.log para ver los registros.
```

sudo python3 ./ssh_brute.py

```
RedTeam@RedTeam:~$ sudo python3 ssh_brute.py
[X] Falló: BlueTeam:1234
[X] Falló: BlueTeam:password
[X] Falló: BlueTeam:admin
[OK] Acceso logrado: BlueTeam:Jb8QHdBy@LDtA8y
```