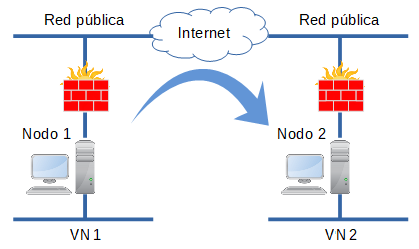
Clase del día - 20/04/2021

El día de hoy vamos a explicar algunos conceptos que serán de utilidad para realizar la tarea 6.

**Red privada y red pública**

Supongamos que creamos dos máquinas virtuales en Azure, cada máquina virtual en un grupo de recursos diferente, esto implica que cada máquina virtual estará conectada a una red virtual (VN) diferente, tal como se muestra en la siguiente figura:



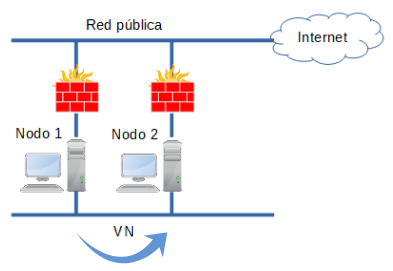
El firewall de una máquina virtual se puede configurar con reglas de entrada y reglas de salida, las reglas de entrada definen qué direcciones públicas y qué puertos se pueden conectar a la máquina virtual, mientras que las reglas de salida definen a qué direcciones públicas y a qué puertos se puede conectar la máquina virtual.

Por seguridad de la máquina virtual, las reglas de entrada suelen ser más restrictivas que las reglas de salida.

En este caso, para que el Nodo-1 se pueda conectar al Nodo-2, solo necesitamos crear una regla de entrada que permita que el Nodo-1 se conecte a través de un puerto específico.

Por otra parte, debido a que las redes virtuales VN1 y VN2 están desconectadas, no es posible conectar el Nodo-1 y el Nodo-2 utilizando las direcciones IP privadas.

Ahora supongamos que **creamos dos máquinas virtuales en el mismo grupo de recursos**. En este caso las dos máquinas virtuales comparten la misma red virtual (VN).



Si el Nodo-1 requiere comunicarse con el Nodo-2 no es necesario crear una regla en el firewall del Nodo-2 ya que ambos nodos están conectados a través de la misma red virtual.

Notar que la comunicación entre las máquinas virtuales mediante la VN se realiza utilizando las direcciones IP privadas de las máquinas virtuales.

**¿Cómo ejecutar Java RMI en la nube?**

Para registrar un objeto remoto en el rmiregistry utilizamos el método rebind().

Debido a que el servidor RMI debe ejecutar en la misma computadora donde ejecuta rmiregistry, la URL que pasa como parámetro al método rebind() deberá incluir el dominio **localhost**, tal como se muestra en el siguiente ejemplo:

public class ServidorRMI

{  
  public static void main(String[] args) throws Exception  
  {  
    String url = "rmi://**localhost**/prueba";  
    ClaseRMI obj = new ClaseRMI();  
  
    // registra la instancia en el rmiregistry  
    Naming.**rebind**(url,obj);  
  }  
}

Para que el cliente RMI pueda invocar los métodos del objeto remoto registrado por el servidor RMI, se debe obtiener una referencia al objeto remoto utilizando el método lookup(). Entonces la URL que pasa como parámetro al método lookup() **deberá definir la IP privada** del nodo dónde ejecuta el servidor RMI.

Supongamos que la dirección IP privada donde ejecuta el servidor RMI, es **10.0.2.4**:

public class ClienteRMI

{  
  public static void main(String args[]) throws Exception  
  {  
    // en este caso el objeto remoto se llama "prueba", notar que se utiliza el puerto default 1099  
    String url = "rmi://**10.0.2.4**/prueba";  
  
    // obtiene una referencia que "apunta" al objeto remoto asociado a la URL  
    InterfaceRMI r = (InterfaceRMI)Naming.lookup(url);  
  
    System.out.println(r.mayusculas("hola"));  
    System.out.println("suma=" + r.suma(10,20));  
  
    int[][] m = {{1,2,3,4},{5,6,7,8},{9,10,11,12}};  
    System.out.println("checksum=" + r.checksum(m));  
  }  
}

Actividades individuales a realizar

1. Crear dos máquinas virtuales (Nodo-1 y Nodo-2) en el mismo grupo de recursos.
2. Compilar el programa [ClienteRMI.java](https://m4gm.com/moodle/mod/resource/view.php?id=1791) en el Nodo-1. Utilizar la IP privada del Nodo-2 en la URL.
3. Compilar el programa [ServidorRMI.java](https://m4gm.com/moodle/mod/resource/view.php?id=1790) en el Nodo-2. Utilizar localhost en la URL.
4. Ejecutar en el Nodo-2: rmiregistry&
5. Ejecutar en el Nodo-2: java ServidorRMI&
6. Ejecutar en el Nodo-1: java ClienteRMI&

**IMPORTANTE: se debe eliminar las máquinas virtuales y todos sus recursos lo más pronto posible, ya que se deberá ahorrar saldo para poder realizar las tareas siguientes.**