Orquestador – Migración a un nuevo servidor

Esta guía describe el procedimiento técnico recomendado para trasladar la operación de un Orquestador Rocketbot existente hacia una nueva infraestructura, asegurando la continuidad de los servicios y la correcta configuración de los activos digitales.

Requisitos Previos

Antes de iniciar el proceso de traslado, es indispensable cumplir con los siguientes puntos:

- Contar con la aprobación formal del área comercial.
- Gestionar una nueva licencia de Orquestador desde el portal oficial de Rocketbot.
- Verificar el acceso administrativo tanto al servidor de origen (actual) como al de destino (nuevo).

Configuración del Nuevo Entorno

Tras la solicitud de la licencia, se habilitará el acceso al nuevo servidor:

- Se proporcionarán las credenciales de acceso para la nueva cuenta.
- La dirección de acceso estándar para el nuevo Orquestador será: https://orchestrator.myrb.io.

Ejecución de la Migración

Es fundamental comprender que la migración no es un proceso automatizado; debe realizarse de forma manual replicando la estructura del entorno anterior.

Elementos a Trasladar

Se debe asegurar la réplica de los siguientes componentes clave:

- **Procesos:** Flujos de trabajo y lógicas de negocio.
- **Instancias:** Configuración de los equipos de ejecución.
- **Formularios y Assets:** Recursos compartidos e interfaces de usuario.
- **Usuarios:** Definición de roles y esquemas de permisos.

Actualización de Robots e Integraciones

Aquellos procesos que consuman servicios del Orquestador (como los módulos Rocketbot Xperience o Assets_NOC) requieren una actualización técnica para operar con el nuevo servidor.

Acciones de Reconfiguración

- **Actualización de URL:** Reemplazar cualquier referencia a la URL antigua por la nueva dirección oficial: https://orchestrator.myrb.io.
- **Generación de Nuevas Credenciales:** Es imperativo crear y sustituir en los robots los siguientes identificadores generados en el nuevo servidor:
    - API Keys y Keys de instancias.
    - Tokens de procesos y de formularios.

- **Actualizar los robots** :
    - Reemplazar credenciales antiguas por las nuevas.
    - Validar que las conexiones funcionen correctamente.

Fase de Validación y Cierre

Para minimizar riesgos, ambos servidores permanecerán activos de manera simultánea durante la fase de transición.

1. **Pruebas de Estabilidad:** Se recomienda ejecutar pruebas completas en el nuevo entorno, validando especialmente los procesos críticos y las conexiones con sistemas externos.
2. **Cese del Servicio Anterior:** Una vez confirmada la estabilidad operativa del nuevo servidor, se debe coordinar formalmente la baja del servidor antiguo tras la validación interna de los responsables.

Recomendaciones Finales de Seguridad

- **Control de Tiempos:** Realizar las tareas de migración en franjas horarias con menor impacto operativo.
- **Resguardo de Datos:** Mantener copias de seguridad de todas las configuraciones críticas durante el proceso.
- **Trazabilidad:** Documentar rigurosamente cada cambio efectuado, especialmente en lo que respecta a nuevas claves y tokens generados.
- **Escalabilidad:** En infraestructuras de gran tamaño, se sugiere realizar el traslado de los recursos de forma escalonada por etapas.

NOC v.2025.12.03

Network Orchestrator Client (NOC)

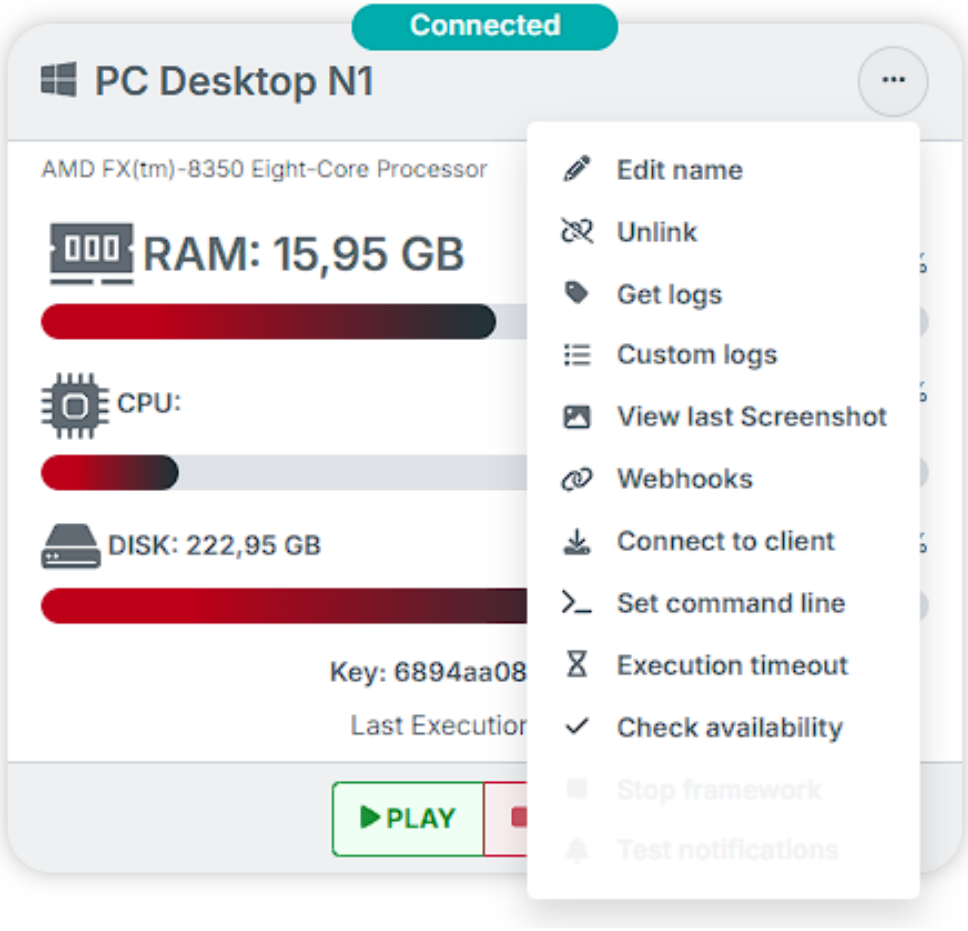
El **Network Orchestrator Client** es la herramienta encargada de vincular la capacidad de ejecución de su equipo local con el servidor del Orquestador. Esta nueva actualización permite visualizar la versión actual del **NOC** e integra herramientas avanzadas para la gestión de reconexiones, notificaciones y modos de ejecución.

Conectar instancia al cliente

De forma predeterminada, asignar un nombre a la instancia no la vincula automáticamente con el equipo. Para establecer la conexión, siga este procedimiento:

1. Haga clic en el menú de la instancia (icono de los tres puntos).
2. Seleccione el botón **[Connect to Client]**.

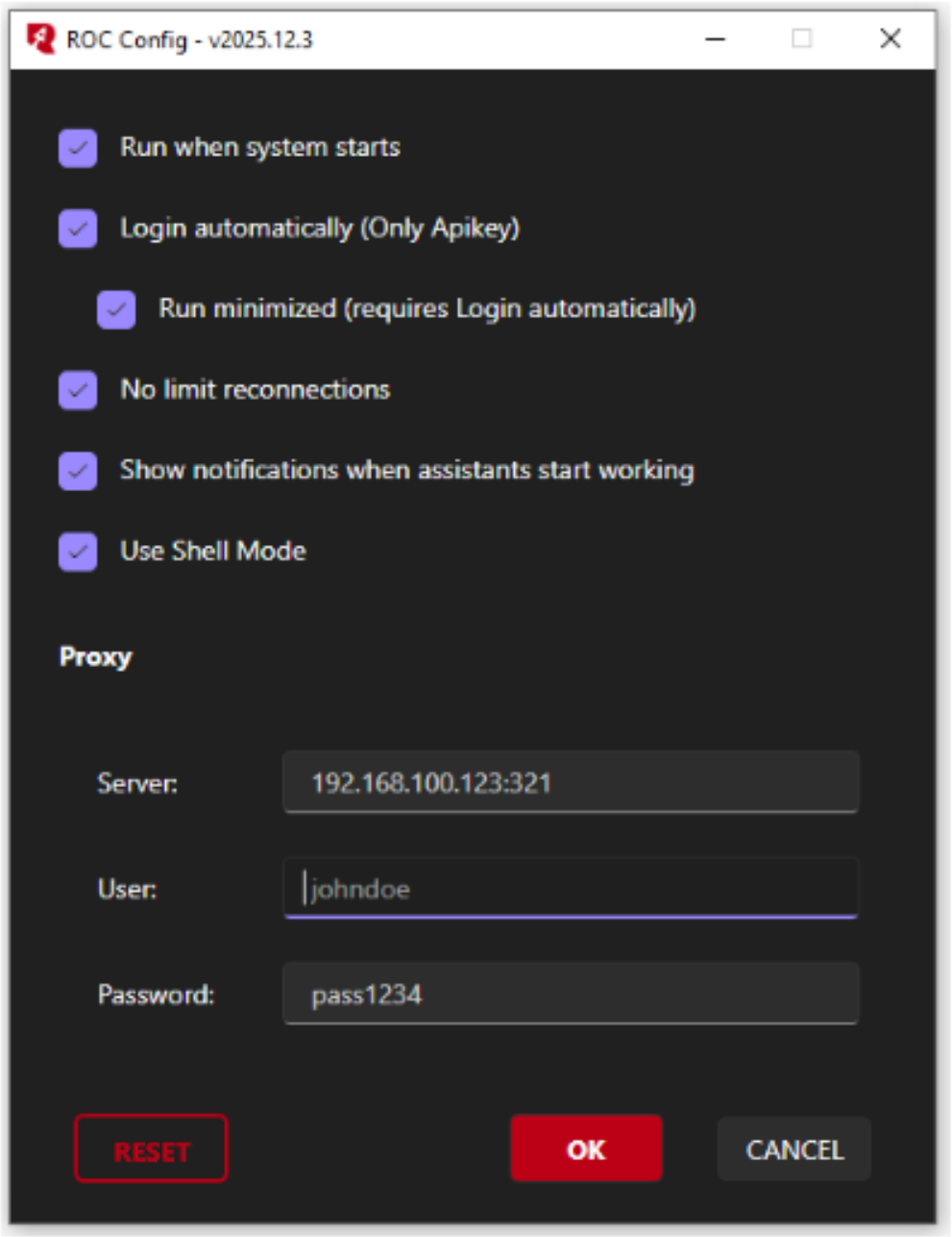
Para una guía detallada paso a paso, consulte la documentación oficial: Configuración de Instancias.



Nueva Actualización y Herramientas

La versión más reciente del cliente (***NOC***)) introduce mejoras significativas en la visibilidad y el control del entorno de ejecución:

- **Visualización de Versión:** Ahora el sistema muestra de forma directa la versión actual del **NOC** instalada en el equipo.
- **No limit reconnections:** Al habilitarse, el cliente intentará restablecer la conexión tras una caída de red de forma indefinida, sin un límite máximo de peticiones.
- **Show notifications when assistants start working:** Esta función muestra una notificación visual con el nombre y la imagen del asistente cuando la instancia ejecuta un robot configurado en modalidad “asistente”.
- **Use Shell Mode:** Permite la ejecución de robots empaquetados en formato .ZIP (consulte la documentación).



Configuración General del Cliente

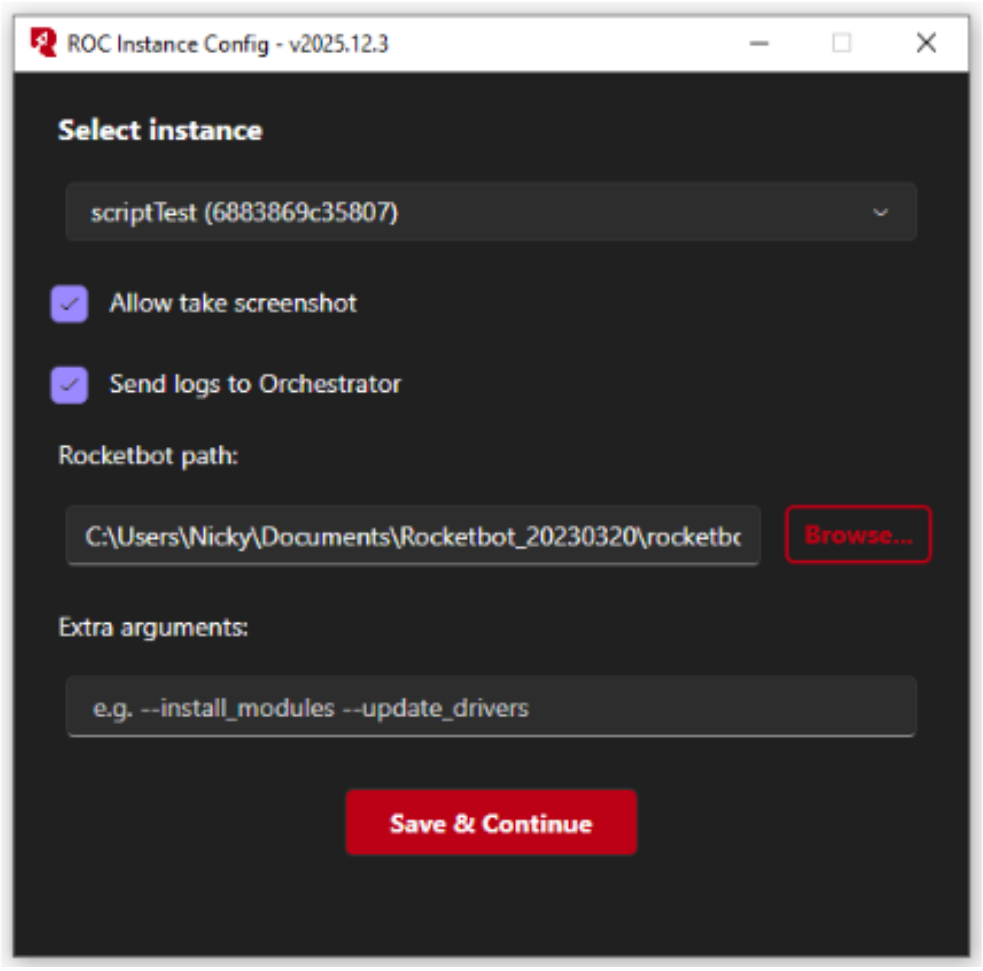
Estas opciones permiten definir el comportamiento inicial del software y las preferencias de conectividad:

Parámetro	Función
Run when Windows starts	Si está habilitado, el cliente se ejecutará automáticamente al iniciar el sistema operativo.
Login automatically (Only API KEY)	Permite iniciar sesión de forma automática al abrir el software si se utiliza una API KEY.
Run minimized	Ejecuta el cliente de forma minimizada en la bandeja del sistema.
No limit reconnections	Ante una pérdida de conexión, el NOC intentará reconectarse de forma indefinida sin límite de peticiones.

Parámetro	Función
Show notifications when assistants start working	Muestra el nombre y la imagen del asistente cuando la instancia ejecuta un robot configurado bajo esa modalidad.
Use Shell Mode	Habilita la utilización de robots en formato tipo .ZIP.
Proxy	Configuración de red para filtros de seguridad. Requiere el protocolo http, seguido de la dirección de salida y el puerto.
Reset	Restablece todos los valores a la configuración original de fábrica.

Parámetros de Instancia

Opciones específicas para la gestión de la instancia seleccionada y el rastro de ejecución:

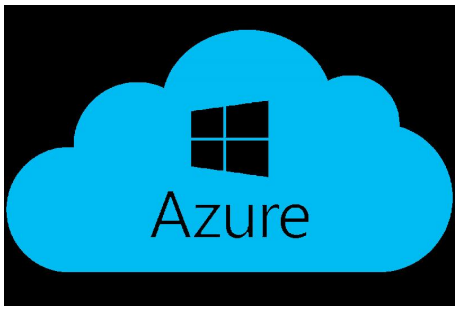


- **Select instance:** Menú desplegable para elegir la instancia que se desea vincular al equipo actual.
- **Allow take screenshot:** Permite capturar imágenes de la pantalla durante la ejecución para verificar procesos o errores visuales.(Ver **screenshot** en herramientas de instancias).
- **Send logs to Orchestrator:** Envía el registro de los comandos ejecutados por el robot hacia el Orquestador para su posterior consulta vía

getlogs.(Ver **“getlogs”** en herramientas de instancias).

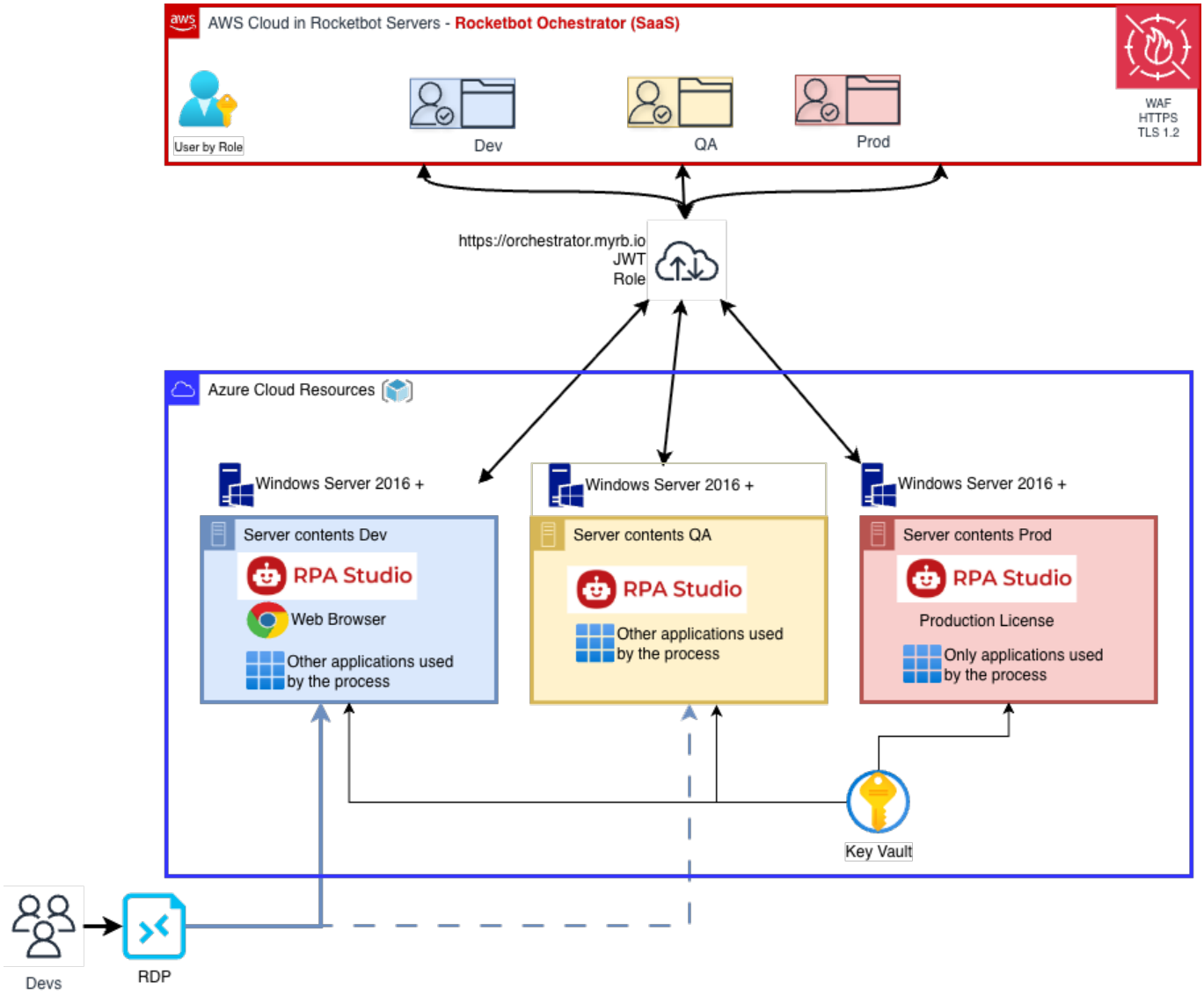
- **Rocketbot path:** Ruta del directorio donde se encuentra el ejecutable de **Rocketbot Studio**.
- **Extra arguments:** Campo para ingresar comandos adicionales, como instalación de módulos o actualización de drivers.

Despliegue de Rocketbot en Azure



En esta guía, te explicaremos los pasos para crear una máquina virtual (VM) en Microsoft Azure y montar Rocketbot Studio para ejecutar robots.

Arquitectura recomendada.



Crear y administrar VM de Windows Server

Creación de una máquina virtual de Windows en Azure Portal

Las máquinas virtuales de Azure pueden crearse mediante Azure Portal. Este método proporciona una interfaz de usuario basada en explorador para crear máquinas virtuales y sus recursos asociados. En esta guía de inicio rápido se muestra cómo usar Azure Portal para implementar una máquina virtual (VM) en Azure que ejecuta Windows Server 2022 Datacenter. Para ver la máquina virtual en acción, conéctese a la máquina virtual mediante RDP e instale al servidor web IIS.

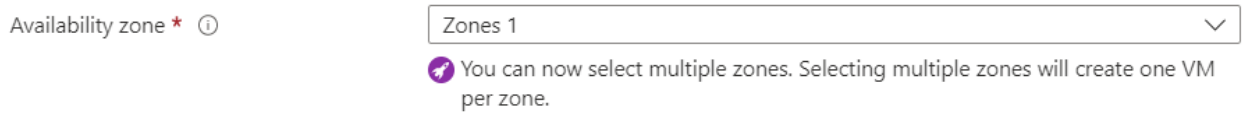
Si no tiene una suscripción a Azure, cree una cuenta gratuita antes de empezar.

Inicio de sesión en Azure

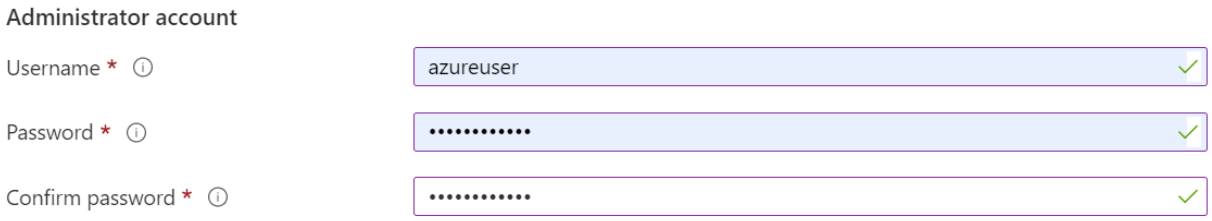
Inicie sesión en Azure Portal.

Crear máquina virtual

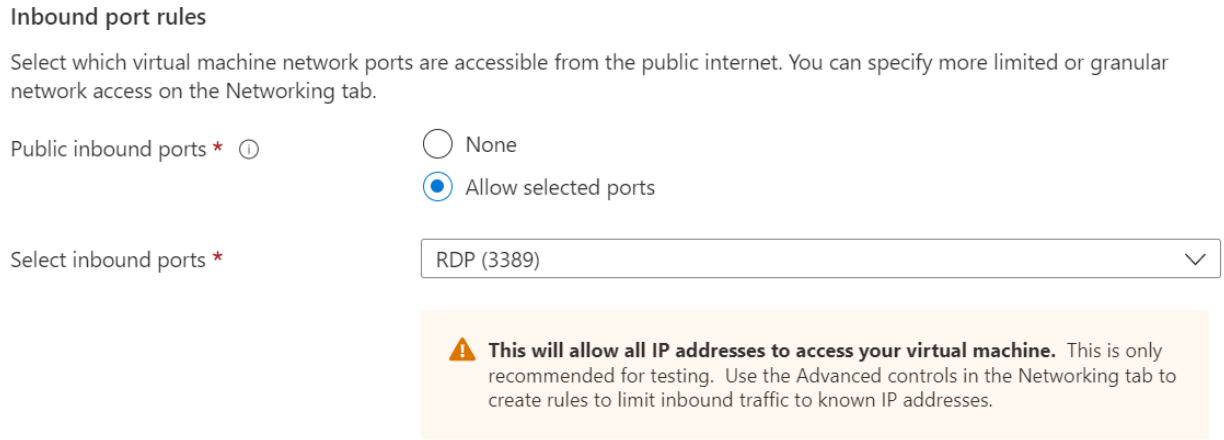
1. Escriba *máquinas virtuales* en el cuadro de búsqueda.
2. En **Servicios**, seleccione **Máquinas virtuales**.
3. En la página **Máquinas virtuales**, selecciona **Crear** y, luego, **Máquina virtual de Azure**. Se abrirá la página **Creación de una máquina virtual**.
4. En **Detalles de instancia**, escriba *myVM* en **Nombre de máquina virtual** y elija *Windows Server 2022 Datacenter: Azure Edition – x64 Gen2* para la **Imagen**. Deje los demás valores predeterminados. Nota:Algunos usuarios ahora verán la opción de crear máquinas virtuales en varias zonas. Para más información sobre esta nueva funcionalidad, consulte Creación de máquinas virtuales en una zona de disponibilidad.



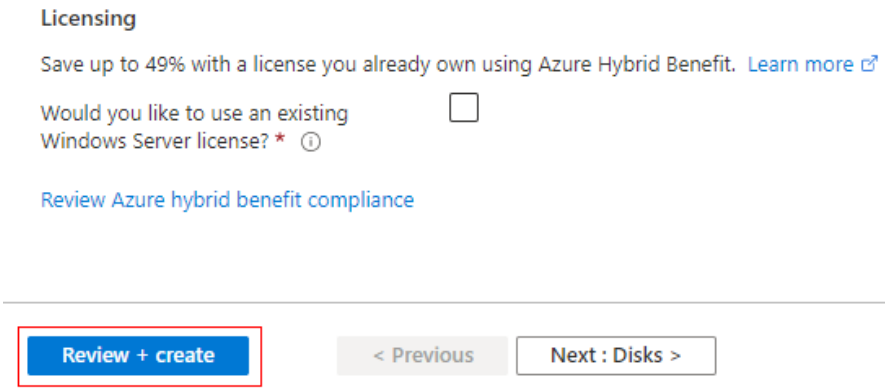
5. En **Cuenta de administrador**, proporcione un nombre de usuario, como *azureuser*, y una contraseña. La contraseña debe tener al menos 12 caracteres de largo y cumplir con los requisitos de complejidad definidos.



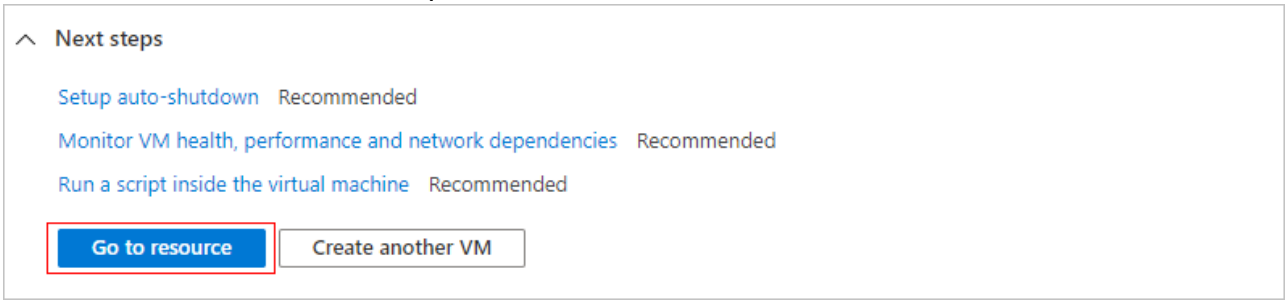
6. En **Reglas de puerto de entrada**, elija **Permitir los puertos seleccionados** y luego seleccione **RDP (3389)** y **HTTP (80)** en la lista desplegable.



7. Deje los valores predeterminados restantes y luego seleccione el botón **Revisar + crear** en la parte inferior de la página.



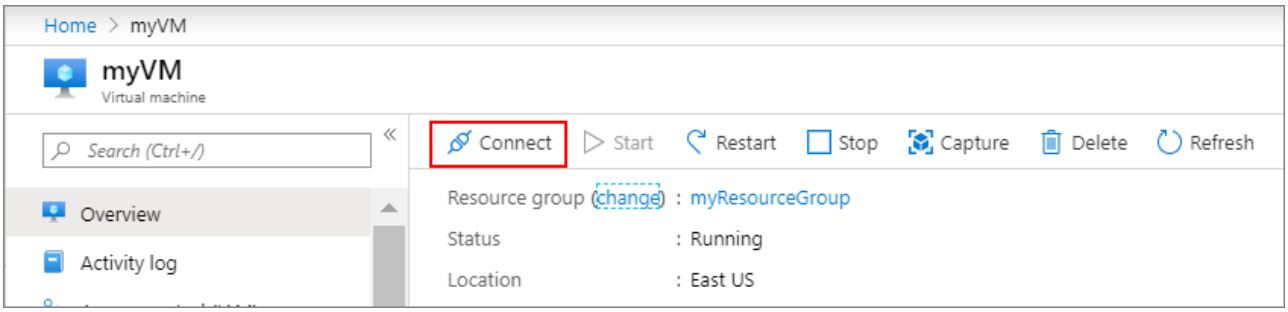
8. Después de que se ejecute la validación, seleccione el botón **Crear** en la parte inferior de la página.
9. Una vez finalizada la implementación, seleccione **Ir al recurso**.



Conexión a la máquina virtual

Cree una conexión a Escritorio remoto en la máquina virtual. Estas instrucciones indican cómo conectarse a la máquina virtual desde un equipo Windows. En un equipo Mac, necesita un cliente RDP como este Cliente de Escritorio remoto de Mac App Store.

1. En la página de información general de la máquina virtual, seleccione **Conectar**>**RDP**.



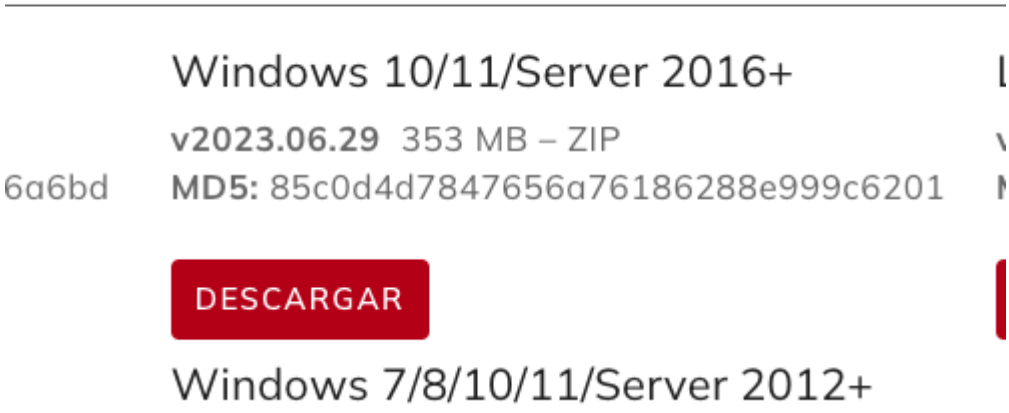
2. En la pestaña **Conectar con RDP**, mantén las opciones predeterminadas para conectarte por la dirección IP, a través del puerto 3389, y haga clic en **Descargar archivo RDP**.
3. Abra el archivo RDP que descargó y haga clic en **Conectar** cuando se le solicite.
4. En la ventana **Seguridad de Windows**, seleccione **Más opciones** y, después, **Usar otra cuenta**. Escribe el nombre de usuario como **localhost***username*, la contraseña que creó para la máquina virtual

y, luego, haz clic en **Aceptar**.
5. Puedes recibir una advertencia de certificado durante el proceso de inicio de sesión. Haga clic en **Sí** o en **Continuar** para crear la conexión.

Fuente: https://learn.microsoft.com/es-es/azure/virtual-machines/windows/quick-create-portal

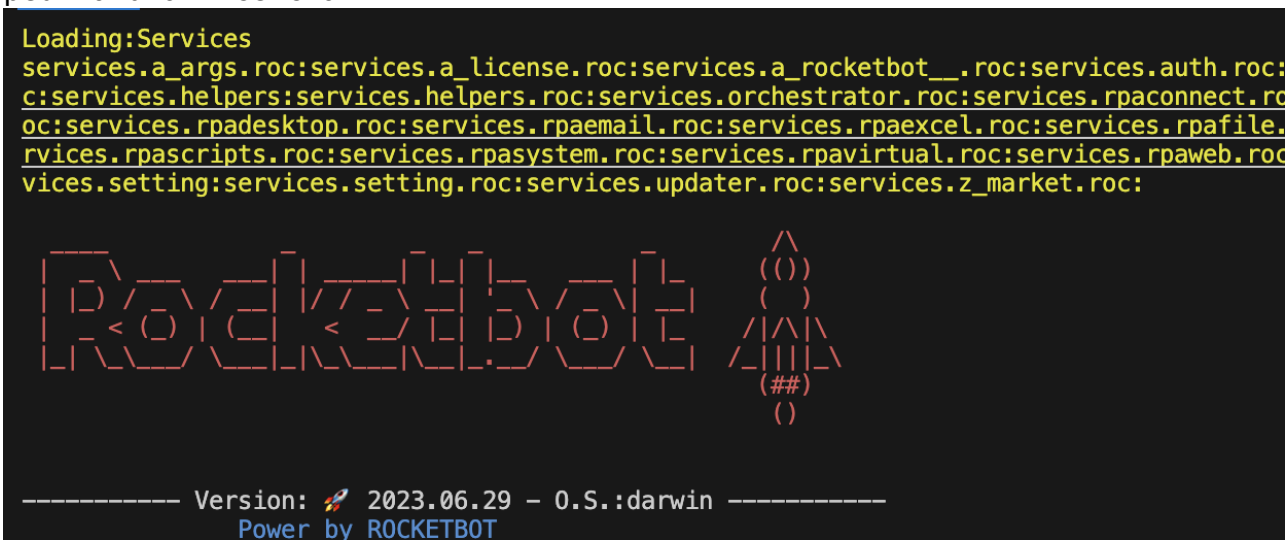
Instalar Rocketbot en Windows

1. Para instalar Rocketbot dirijase a https://rocketbot.com/es/rocketbot-studio-rpa/
2. Elija descargar y seleccione la version de Rocketbot para Windows.
3. Descargue y descomprima el ZIP en un carpeta dentro de la maquina donde ejecutará Rocketbot. por ejemplo: C:\\rocketbot

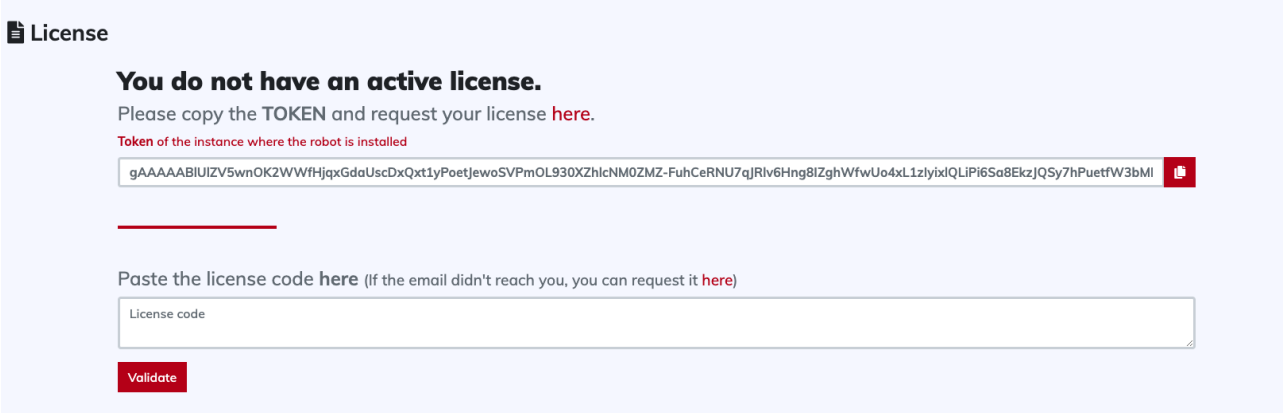


4. Ingrese a la carpeta de Rocketbot y ejecute la aplicacion **rocketbot.exe**

5. Se abrira el navegador por defecto en el puerto 5000 (configurable) y le pedirá una licencia.



6. Copie el Token y complete los datos en la pagina de licencias



7. Recibirá en su correo la licencia. Pegue el código en Rocketbot Studio y comience a programar robots.

Consideraciones y TIPs

Rocketbot Studio/Runner requiere una sesion activa y desktop activo.

Mantener interfaz al desconectarse del servidor

Para evitar estos problemas, los servidores windows cuentan con un comando llamado **tscon.exe** que permite tomar la sesión que tienes abierta en el servidor y moverla a la consola, de esta manera tu robot tendrá una interfaz donde trabajar luego de desconectarte del servidor.

En las pestañas *Desconectarme y ejecutar una tarea* o *Desconectarme y ejecutar ROC* de la aplicación **RDP-Rocketbot** puedes crear un archivo bat con el comando tscon. Este archivo debes ejecutarlo como administrador cada vez que quieras desconectarte del servidor (o ejecutar tu robot, según la opción elegida) y te desconectará del servidor y mantendrá la interfaz gráfica.

Cuidado! Al no abrir el escritorio remoto desde una pantalla física, la resolución puede cambiar y ser menor. Revisa la sección tres para ver como trabajar con esto.

El bat puedes editarlo y revisar que es lo que contiene, pero resumimos lo importante acá. El bat tiene un script de powershell que obtiene el id de la sesión que estas usando al entrar al rdp y la utiliza en el comando tscon para que este la abra en la sesión de la consola.

```
@powershell -NoProfile -ExecutionPolicy unrestricted -Command
"$sessionid=((quser $env:USERNAME | select -Skip 1) -split '\\s+')[2]; tscon
$sessionid /dest:console" 2> UnlockErrors.log
```

Si necesitas que el bat ejecute tu robot al desconectarte, verás algo así

```
@powershell -NoProfile -ExecutionPolicy unrestricted -Command
"$sessionid=((quser $env:USERNAME | select -Skip 1) -split '\\s+')[2]; tscon
$sessionid /dest:console" 2> UnlockErrors.log
```

```
timeout 10
cd c:\\rocketbot
rocketbot.exe -start=nombre_robot
```

Mantener la interfaz al minimizar el escritorio remoto

Para poder minimizar el escritorio remoto y no afectar el funcionamiento del robot, debes modificar algunos registros en la máquina que abre el rdp, estos registros son:

- Para el usuario actual:
  - HKEY_CURRENT_USER\\Software\\Microsoft\\Terminal Server Client (32 bit)
  - HKEY_CURRENT_USER\\Software\\Wow6432Node\\Microsoft\\Terminal Server Client (64 bit)

- Para todos los usuarios:
  - HKEY_LOCAL_MACHINE\\Software\\Microsoft\\Terminal Server Client (32 bit)
  - HKEY_LOCAL_MACHINE\\Software\\Wow6432Node\\Microsoft\\Terminal Server Client (64 bit)

En la pestaña *Conectarme a Escritorio Remoto* de la aplicación RDP-Rocketbot, puedes cambiar los registros con un solo botón. Solo recuerda ejecutar la aplicación como administrador.

Cambio de resolución al desconectarte del servidor

Si ya solucionaste los problemas anteriores, pero tu **robot** falla y/o aún no reconoce las imagenes, es probable que tengas un problema con la **resolución**. Ya mencionamos en la primera parte de este post que al desconectarte el servidor pierde la interfaz gráfica y es necesario utilizar **tscon**. Esto permite que el robot pueda sacar capturas, pero como el servidor ya no cuenta con tu pantalla para ajustar una resolución, toma sus propias resoluciones por defecto, que en la mayoría de los casos es de 1024×768.

Si tu robot trabaja con **virtualización**, este cambio de resolución puede afectarlo, por lo cual es importante que, antes de comenzar a desarrollar tu robot, revises las resoluciones que permite tu servidor al desconectarte.

En la pestaña *Ver las resoluciones de pantalla* de la aplicación RDP-Rocektbot, puedes ver las resoluciones que permite el servidor en la tabla *Resolucoones de pantalla desde la consola de TSCON*.

Esas resoluciones son las resoluciones que puedes utilizar, pero no quiere decir que puedes comenzar a construir el robot. Debes revisar la resolución que tiene el servidor al desconectarte y si es posible cambia a la resolución que necesitas. Para cambiar la **resolución de pantalla**, puedes utilizar el módulo Windows y también puedes utilizar el comando *Sacar captura de pantall*a de la sección Desktop para válidar si la resolución es la correcta o no.

Ésto lo puedes realizar con **RDP-Helper**, acá encontrarás los pasos necesarios Pasos Resolución

Si al salir de un escritorio remoto no es posible modificar la resolución de forma manual y/o usando el módulo de Rocketbot, puedes modificar realizar la siguiente modificación:

Buscar en el regedit todos los registros que se llamen ***DefaultSettings_XResolution*** y ***DefaultSettings_YResolution*** y cambiar la resolución a la que necesitas

Prevenir el bloqueo de pantalla

Para evitar que el servidor bloquee la pantalla debes, en primer lugar, consultar con el equipo de TI si es posible modificar las configuraciones necesarias para que esto no ocurra, ya que dependiendo de las politicas de seguridad de la empresa, podría no ser posible modificar esta configuración.

En la sección *Configuraciones* de la aplicación RDP-Rocketbot, puedes modificar los registros necesarios para deshabilitar el bloqueo de pantalla. El registro que se modificará es *HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows*

Cambiando Policies Link

For Windows 7/Windows 10/Windows Server 2012 and above:

- Press + and type ***gpedit.msc***Windows+R
- Navigate the tree view on the left to ***Computer Configuration/Administrative Templates/Windows Components/Remote Desktop Services/Remote Desktop Session Host/Session Time Limits***
- Adjust the four settings as desired (I believe you want to adjust ***Set time limit for disconnected sessions*** and ***Terminate session when time limits are reached***)

Habilitar Pantallas Virtuales

En algunos casos, la VM solo permitía una resolución muy baja por defecto. Siguiendo la documentación indicada y asegurándonos de **apagar la VM antes de activar la casilla de configuración**, logramos establecer una resolución adecuada para el proceso requerido.

Conectar con Orquestador

Para conectar con el Orquestador de Rocketbot debe contar con una cuenta habilitada de Rocketbot Orchestrator Center. En la VM de AWS tiene que tener habilitado el acceso a las web ***.myrb.io** y ***.rocketbot.com** esto permitirá al cliente de orquestador NOC conectarse con las url de api de orquestador.

Permisos de red y dominios

Para la conexión con Orquestador debe estar habilitado el acceso a:

- ***.myrb.io**
- **Puerto: 443 (HTTPS)**

El puerto 443 debe estar **libre** y permitido para conexiones salientes sin inspección profunda (SSL inspection).

Conectividad y rendimiento

Para evitar timeouts y desconexiones durante ejecuciones o conexiones con Orquestador, se requieren:

- **Tiempo de respuesta máximo recomendado (timeout): ≤ 30 segundos**
- **Ancho de banda mínimo recomendado: 200–250 Mbps**

Habilitar URL

- https://storage.googleapis.com/* (URL de descarga de los drivers del navegador, en caso de que automatices Chrome)
- https://chromedriver.storage.googleapis.com/* (URL para mantener sincronizado el webdriver con la versión de Google Chrome)
- https://googlechromelabs.github.io/* (URL de revisión y obtención de la versión de los drivers de Chrome)
- https://market-api.rocketbot.co/* (URL de descarga y actualización de módulos de Rocketbot)
- http://raw.githubusercontent.com/* (URL de consulta para actualizaciones de Rocketbot Studio)

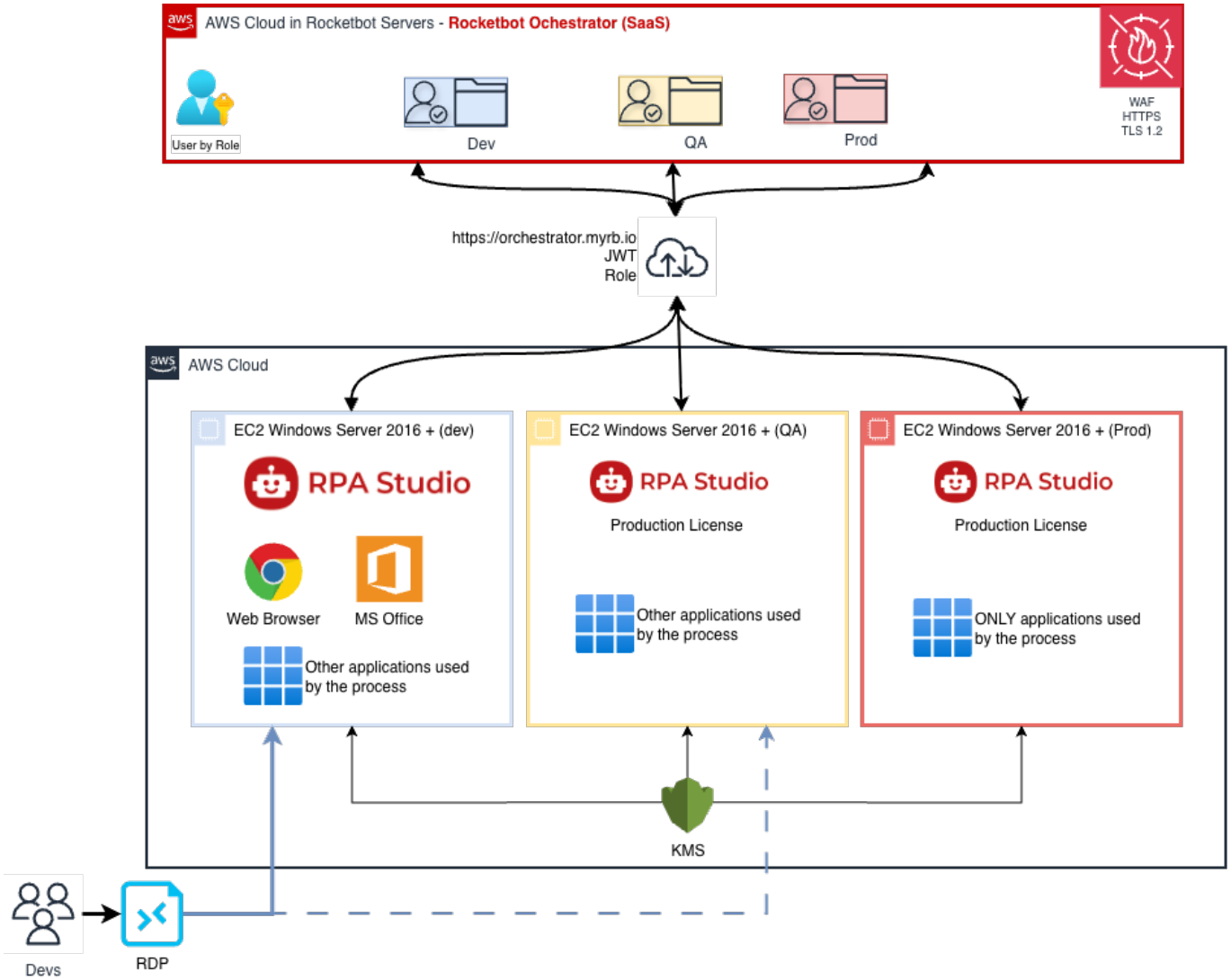
INSTALACIÓN Y CONFIGURACIÓN CLIENTE ORQUESTADOR

Para conectar una instacia siga la siguiente GUIA: https://docs.rocketbot.com/2024/08/23/orquestador-rocketbot-instancias/#Conectar_instancia_al_cliente

Despliegue de Rocketbot en AWS

En esta guía, te explicaremos los pasos para crear una máquina virtual (VM) en AWS Cloud (AWS) y montar Rocketbot Studio para ejecutar robots.

Arquitectura recomendada.



Crear y administrar VM de Windows Server

Antes de comenzar revise los requerimientos minimos aqui: https://docs.rocketbot.com/2024/05/15/requerimientos-de-hardware-y-software/

Crear VM de Windows Server

Para crear una instancia de Windows vamos **Instancias –- Launch Instance.**

Comenzará el asistente pidiendo que escojamos el AMI, donde en la barra de búsqueda escribiremos Windows para que nos filtre la búsqueda.

Nos hará la búsqueda y seleccionamos la imagen de Windows que necesitamos o queremos.

Recomendada Windows Server 2016 o Superior

Si se quiere usar la capa gratuita, seleccionamos el tipo de Instancia gratuita, t2.micro.

Se recomienda minimo t2.medium

A partir de aquí dejare toda la configuración por defecto, ya que el propósito de este laboratorio es poder crear y conectarme a una instancia de Windows.

En tamaño de disco lo dejamos en 30 GB y siguiente.

Revisión de la configuración y launch para lanzar la creación de la instancia.

Antes de Lanzar la creación de la instancia, no pedirá seleccionar una key pair existente o crear una nueva. Como previamente he creado una nueva, la selecciono y lanzo la instancia.

Comenzará la creación.

Si vamos al consola veremos como se va inicializando la instancia.

Conectar con la instancia creada

Desplegada la instancia, si intentamos conectarnos a la instancia desde la consola

Nos avisará que tenemos que esperar al menos 4 minutos desde el despliegue de la instancia.

Una vez pasado el tiempo intentamos de nuevo conectarnos y nos aparecerá la ventana de conectar a la instancia, donde haremos clic en el botón **Download Remote Desktop File**.

Nos descargará el acceso RDP a la instancia.

Obtener credenciales

Si hacemos clic en el botón **Get Password**

Nos dirá de seleccionar la key pair con la que hemos configurado al crear la instancia, la seleccionamos y hacemos clic en **Decrypt Password.**

Nos generará el password del usuario Administrator de la Instancia y lo copiaremos en el portapapeles.

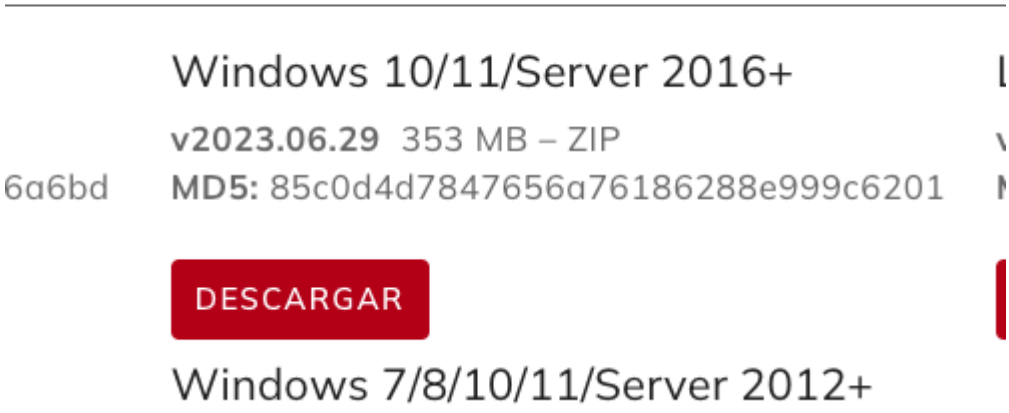
Copiamos el password y nos conectamos a la instancia desde el acceso RDP descargado, introducimos el password y ya podremos conectarnos a la instancia. Una vez conectados, lo primero que tendremos que hacer es cambiar el password.

Como ultimo apunte, también podemos coger el password de Windows desde **Actions – Get Windows Password.**

Fuente: https://docs.aws.amazon.com/es_es/AWSEC2/latest/WindowsGuide/ec2-windows-instances.html

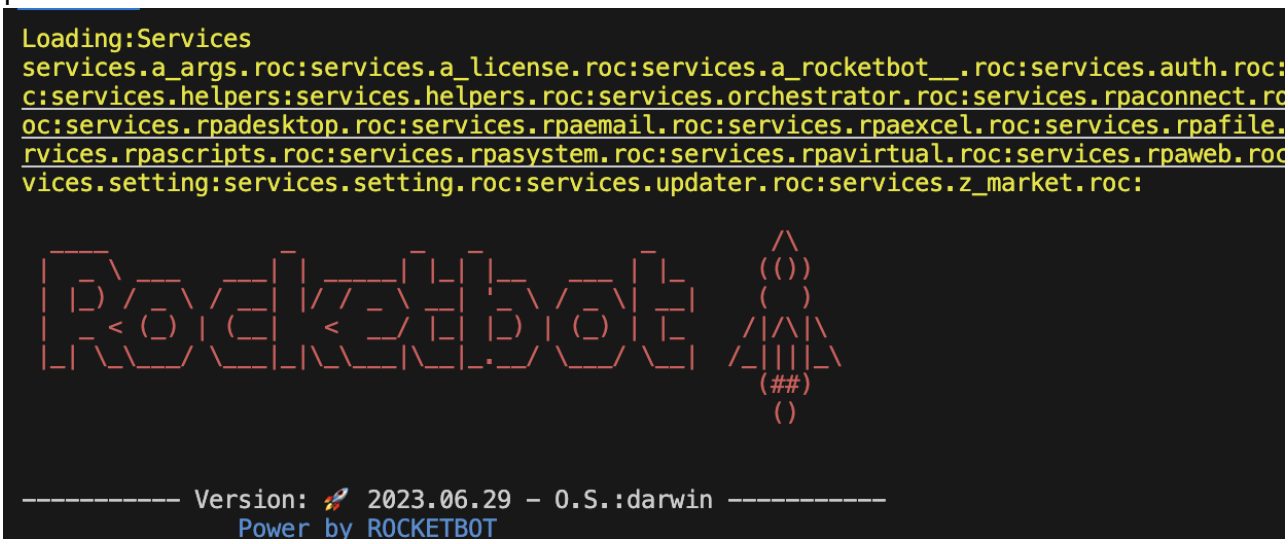
Instalar Rocketbot en Windows

1. Para instalar Rocketbot dirijase a https://rocketbot.com/es/rocketbot-studio-rpa/
2. Elija descargar y seleccione la version de Rocketbot para Windows.
3. Descargue y descomprima el ZIP en un carpeta dentro de la maquina donde ejecutará Rocketbot. por ejemplo: C:\\rocketbot

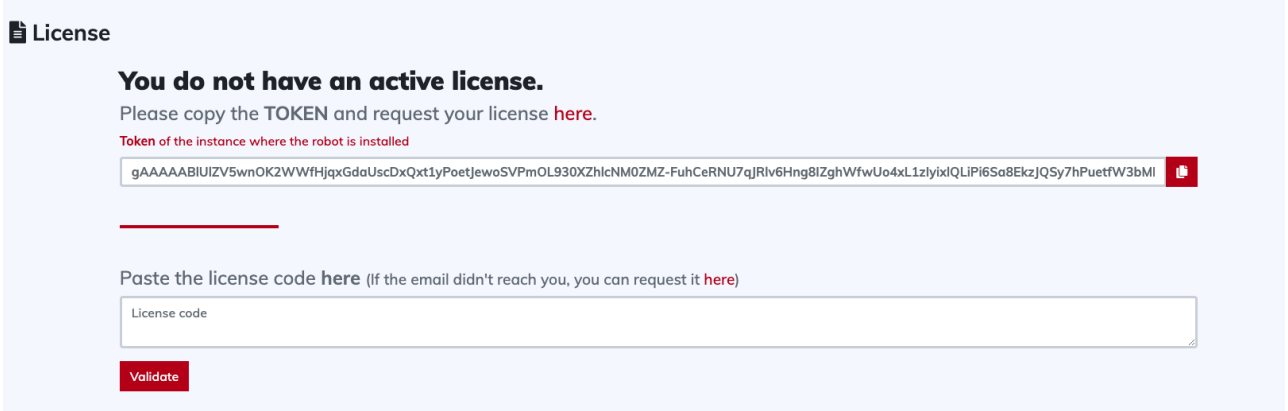


4. Ingrese a la carpeta de Rocketbot y ejecute la aplicacion **rocketbot.exe**

5. Se abrira el navegador por defecto en el puerto 5000 (configurable) y le pedirá una licencia.



6. Copie el Token y complete los datos en la pagina de licencias



7. Recibirá en su correo la licencia. Pegue el código en Rocketbot Studio y comience a programar robots.

Consideraciones y TIPs

Rocketbot Studio/Runner requiere una sesion activa y desktop activo.

Mantener interfaz al desconectarse del servidor

Para evitar estos problemas, los servidores windows cuentan con un comando llamado **tscon.exe** que permite tomar la sesión que tienes abierta en el servidor y moverla a la consola, de esta manera tu robot tendrá una interfaz donde trabajar luego de desconectarte del servidor.

En las pestañas *Desconectarme y ejecutar una tarea* o *Desconectarme y ejecutar ROC* de la aplicación **RDP-Rocketbot** puedes crear un archivo bat con el comando tscon. Este archivo debes ejecutarlo como administrador cada vez que quieras desconectarte del servidor (o ejecutar tu robot, según la opción elegida) y te desconectará del servidor y mantendrá la interfaz gráfica.

Cuidado! Al no abrir el escritorio remoto desde una pantalla física, la resolución puede cambiar y ser menor. Revisa la sección tres para ver como trabajar con esto.

El bat puedes editarlo y revisar que es lo que contiene, pero resumimos lo importante acá. El bat tiene un script de powershell que obtiene el id de la sesión que estas usando al entrar al rdp y la utiliza en el comando tscon para que este la abra en la sesión de la consola.

```
@powershell -NoProfile -ExecutionPolicy unrestricted -Command
"$sessionid=((quser $env:USERNAME | select -Skip 1) -split '\\s+')[2]; tscon
$sessionid /dest:console" 2> UnlockErrors.log
```

Si necesitas que el bat ejecute tu robot al desconectarte, verás algo así

```
@powershell -NoProfile -ExecutionPolicy unrestricted -Command
"$sessionid=((quser $env:USERNAME | select -Skip 1) -split '\\s+')[2]; tscon
$sessionid /dest:console" 2> UnlockErrors.log

timeout 10
cd c:\\rocketbot
rocketbot.exe -start=nombre_robot
```

Mantener la interfaz al minimizar el escritorio remoto

Para poder minimizar el escritorio remoto y no afectar el funcionamiento del robot, debes modificar algunos registros en la máquina que abre el rdp, estos registros son:

- Para el usuario actual:
  - HKEY_CURRENT_USER\\Software\\Microsoft\\Terminal Server Client (32 bit)
  - HKEY_CURRENT_USER\\Software\\Wow6432Node\\Microsoft\\Terminal Server Client (64 bit)

- Para todos los usuarios:
  - HKEY_LOCAL_MACHINE\\Software\\Microsoft\\Terminal Server Client (32 bit)
  - HKEY_LOCAL_MACHINE\\Software\\Wow6432Node\\Microsoft\\Terminal Server Client (64 bit)

En la pestaña *Conectarme a Escritorio Remoto* de la aplicación RDP-Rocketbot, puedes cambiar los registros con un solo botón. Solo recuerda ejecutar la aplicación como administrador.

Cambio de resolución al desconectarte del servidor

Si ya solucionaste los problemas anteriores, pero tu **robot** falla y/o aún no reconoce las imagenes, es probable que tengas un problema con la **resolución**. Ya mencionamos en la primera parte de este post que al desconectarte el servidor pierde la interfaz gráfica y es necesario utilizar **tscon**. Esto permite que el robot pueda sacar capturas, pero como el servidor ya no cuenta con tu pantalla para ajustar una resolución, toma sus propias resoluciones por defecto, que en la mayoría de los casos es de 1024×768.

Si tu robot trabaja con **virtualización**, este cambio de resolución puede afectarlo, por lo cual es importante que, antes de comenzar a desarrollar tu robot, revises las resoluciones que permite tu servidor al desconectarte.

En la pestaña *Ver las resoluciones de pantalla* de la aplicación RDP-Rocektbot, puedes ver las resoluciones que permite el servidor en la tabla *Resolucoones de pantalla desde la consola de TSCON*.

Esas resoluciones son las resoluciones que puedes utilizar, pero no quiere decir que puedes comenzar a construir el robot. Debes revisar la resolución que tiene el servidor al desconectarte y si es posible cambia a la resolución que necesitas. Para cambiar la **resolución de pantalla**, puedes utilizar el módulo Windows y también puedes utilizar el comando *Sacar captura de pantall*a de la sección Desktop para válidar si la resolución es la correcta o no.

Ésto lo puedes realizar con **RDP-Helper**, acá encontrarás los pasos necesarios Pasos Resolución

Si al salir de un escritorio remoto no es posible modificar la resolución de forma manual y/o usando el módulo de Rocketbot, puedes modificar realizar la siguiente modificación:

Buscar en el regedit todos los registros que se llamen ***DefaultSettings_XResolution*** y ***DefaultSettings_YResolution*** y cambiar la resolución a la que necesitas

Prevenir el bloqueo de pantalla

Para evitar que el servidor bloquee la pantalla debes, en primer lugar, consultar con el equipo de TI si es posible modificar las configuraciones necesarias para que esto no ocurra, ya que dependiendo de las politicas de seguridad de la empresa, podría no ser posible modificar esta configuración.

En la sección *Configuraciones* de la aplicación RDP-Rocketbot, puedes modificar los registros necesarios para deshabilitar el bloqueo de pantalla. El registro que se modificará es *HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows*

Cambiando Policies Link

For Windows 7/Windows 10/Windows Server 2012 and above:

- Press + and type ***gpedit.msc***Windows+R
- Navigate the tree view on the left to ***Computer Configuration/Administrative Templates/Windows Components/Remote Desktop Services/Remote Desktop Session Host/Session Time Limits***
- Adjust the four settings as desired (I believe you want to adjust ***Set time limit for disconnected sessions*** and ***Terminate session when time limits are reached***)

Habilitar Pantallas Virtuales

En algunos casos, la VM solo permitía una resolución muy baja por defecto. Siguiendo la documentación indicada y asegurándonos de **apagar la VM antes de activar la casilla de configuración**, logramos establecer una resolución adecuada para el proceso requerido.

Conectar con Orquestador

Para conectar con el Orquestador de Rocketbot debe contar con una cuenta habilitada de Rocketbot Orchestrator Center. En la VM de AWS tiene que tener habilitado el acceso a las web ***.myrb.io** y ***.rocketbot.com** esto permitirá al cliente de orquestador NOC conectarse con las url de api de orquestador.

Permisos de red y dominios

Para la conexión con Orquestador debe estar habilitado el acceso a:

- ***.myrb.io**
- **Puerto: 443 (HTTPS)**

El puerto 443 debe estar **libre** y permitido para conexiones salientes sin inspección profunda (SSL inspection).

Conectividad y rendimiento

Para evitar timeouts y desconexiones durante ejecuciones o conexiones con Orquestador, se requieren:

- **Tiempo de respuesta máximo recomendado (timeout): ≤ 30 segundos**
- **Ancho de banda mínimo recomendado: 200–250 Mbps**

Habilitar URL

- https://storage.googleapis.com/* (URL de descarga de los drivers del navegador, en caso de que automatices Chrome)
- https://chromedriver.storage.googleapis.com/* (URL para mantener sincronizado el webdriver con la versión de Google Chrome)
- https://googlechromelabs.github.io/* (URL de revisión y obtención de la versión de los drivers de Chrome)
- https://market-api.rocketbot.co/* (URL de descarga y actualización de módulos de Rocketbot)
- http://raw.githubusercontent.com/* (URL de consulta para actualizaciones de Rocketbot Studio)

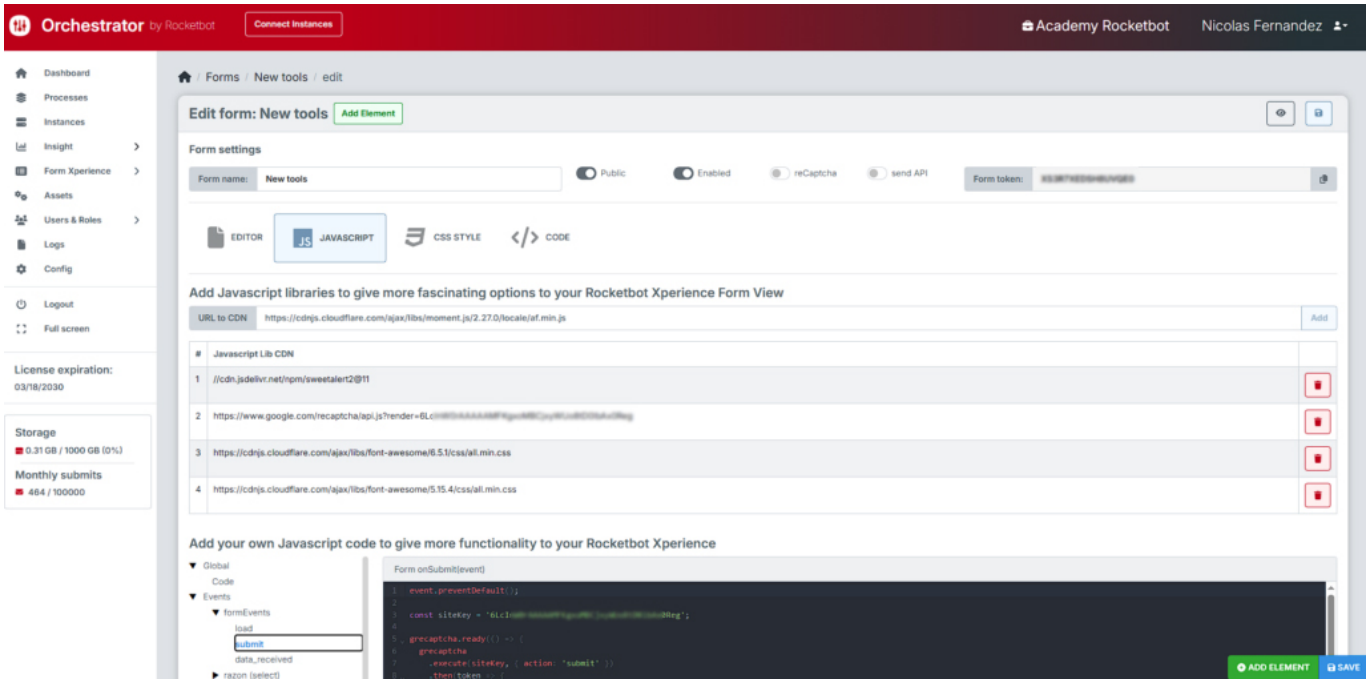
INSTALACIÓN Y CONFIGURACIÓN CLIENTE ORQUESTADOR

Para conectar una instacia siga la siguiente GUIA: https://docs.rocketbot.com/2024/08/23/orquestador-rocketbot-instancias/#Conectar_instancia_al_cliente

---

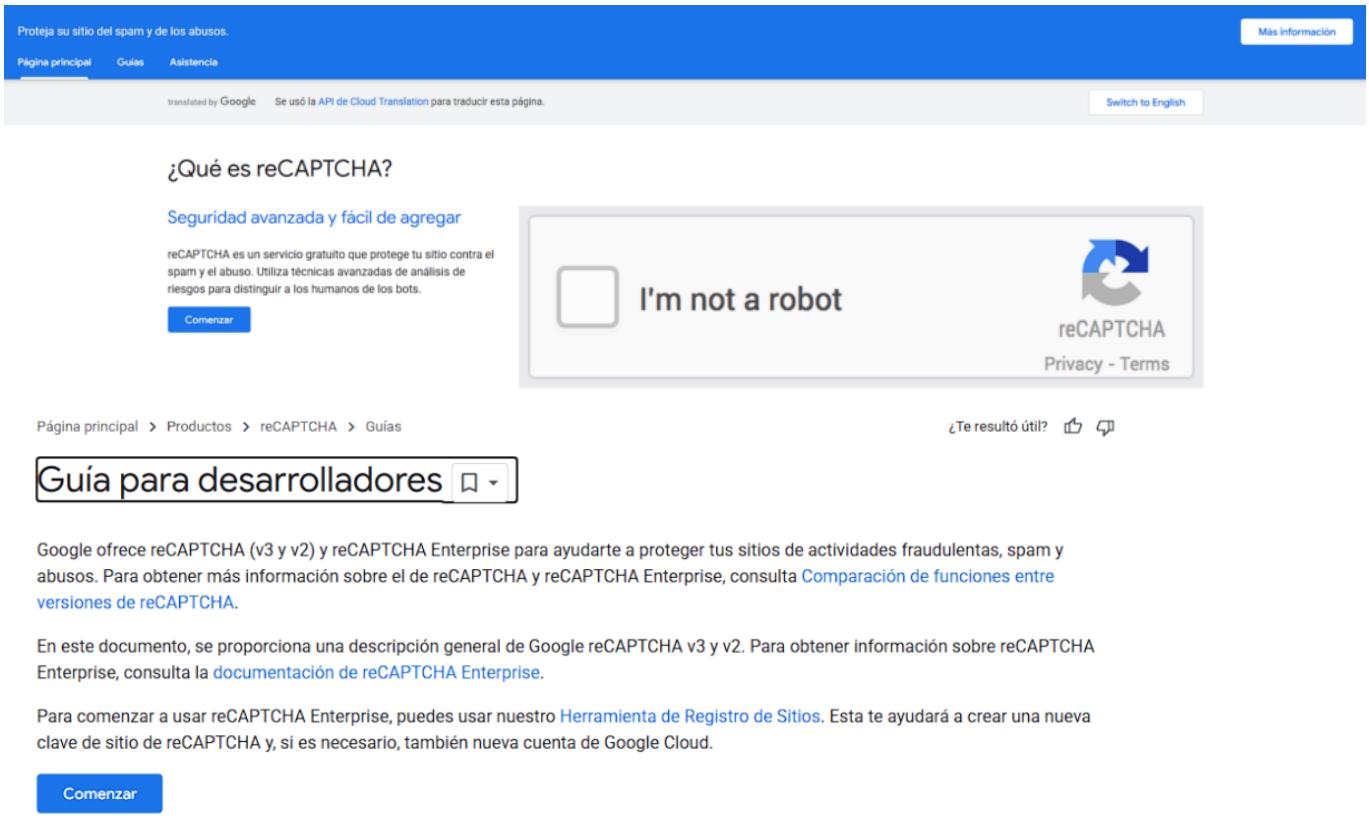
Activar Re-captcha en los formularios

1° Ir al editor del formulario en la seccion JS.

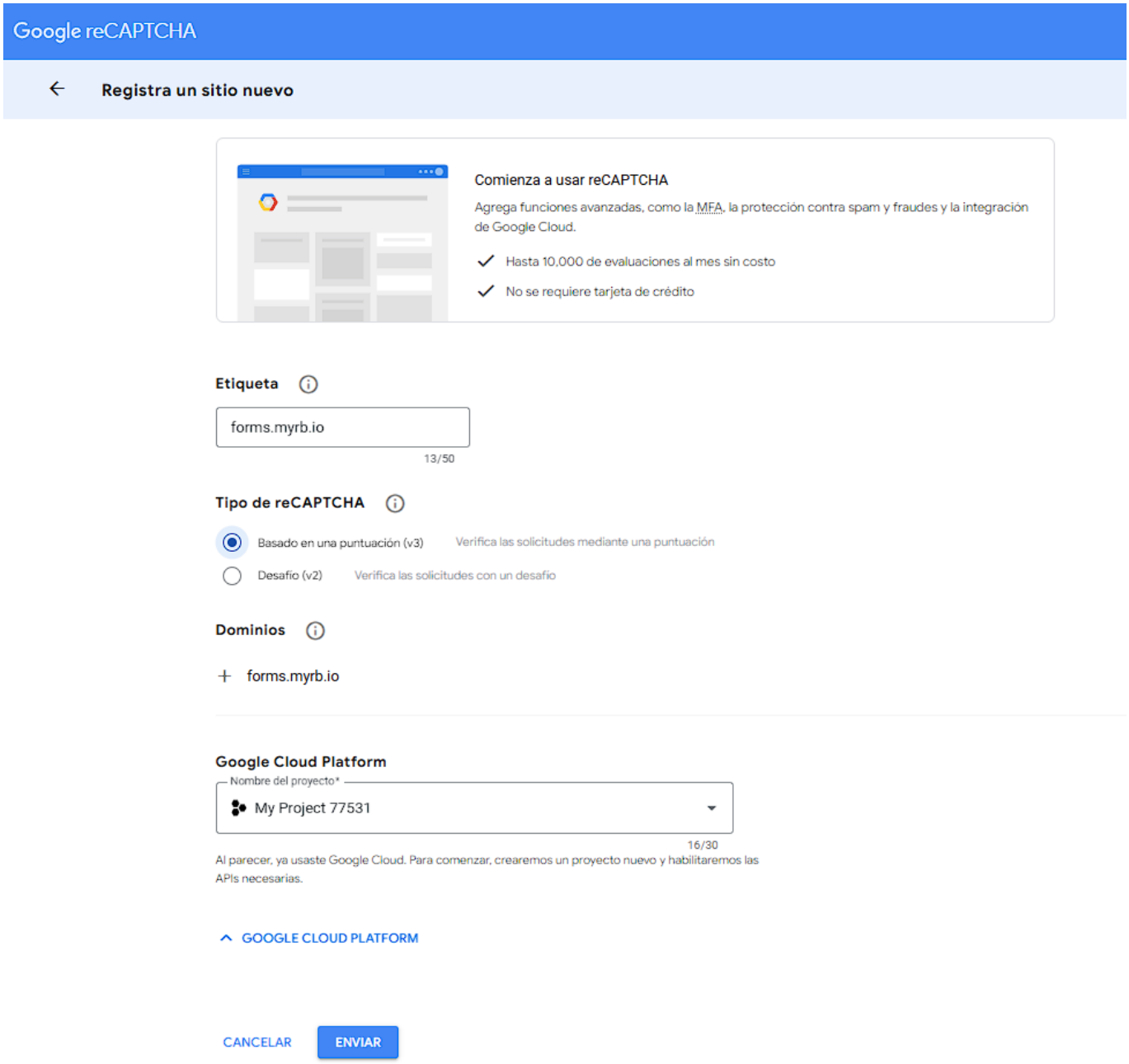


2° Crear la “clave recaptcha” en la pagina de recaptcha google(https://developers.google.com/recaptcha?hl=es-419).

3° Ir a “comenzar”, nos llevará a la guía para desarrolladores, y ahi tambien daremos en “comenzar”.



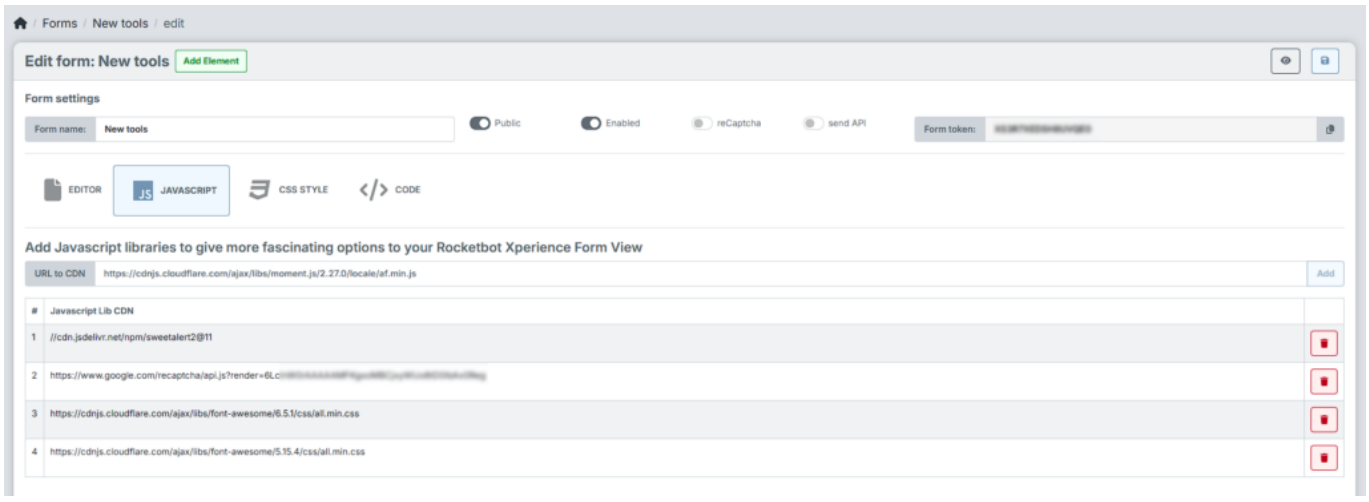
4° Registrar el formulario que quieres utilizar. (En dominio y en etiqueta coloque “forms.myrb.io”.



5° Configurar v3 (Verifica las solicitudes mediante una puntuación) o V2 (Verifica las solicitudes con un desafío).

Luego ir a “enviar” y nos redirigira a nuestra nueva “clave de sitio”.

6° En nuestro formumario ir a la pestaña “js” , agregar un nuevo CDN con “https://www.google.com/recaptcha/api.js?render= + nuestra clave de sitio”

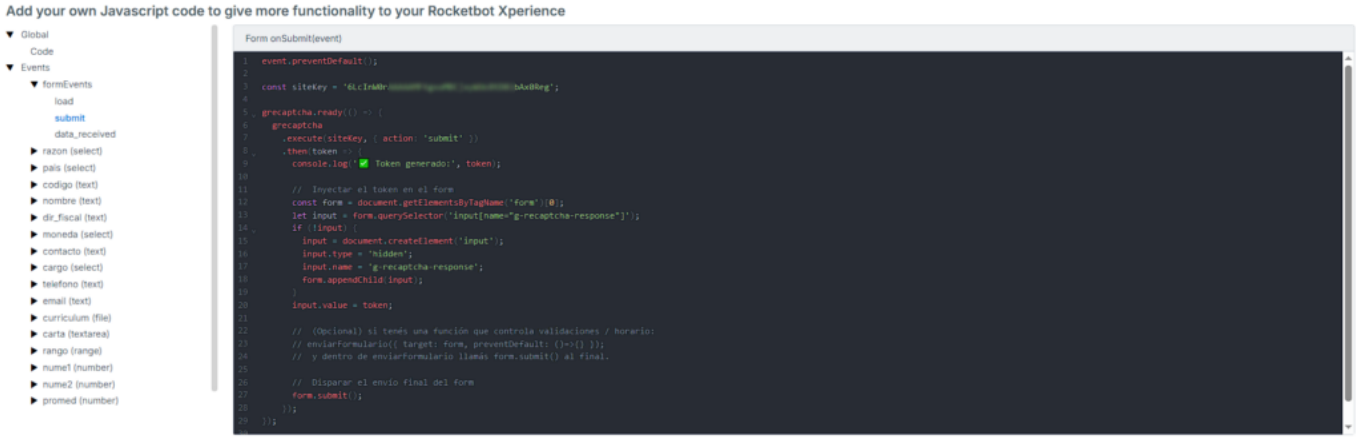


7° En la terminal js -> Events/FormEvents/load tendremos que cargar el recapcha:

```
const siteKey = ‘nuestroCodigoSiteKey’;
// Verificamos si ya se cargó el script
if (typeof grecaptcha === ‘undefined’) {
const script = document.createElement(‘script’);
script.src = `https://www.google.com/recaptcha/api.js?render=${siteKey}`;
script.async = true;
script.defer = true;
document.head.appendChild(script);
}
```

8° En la terminal js -> Events/FormEvents/submit tendremos que ejecutar el recapcha:

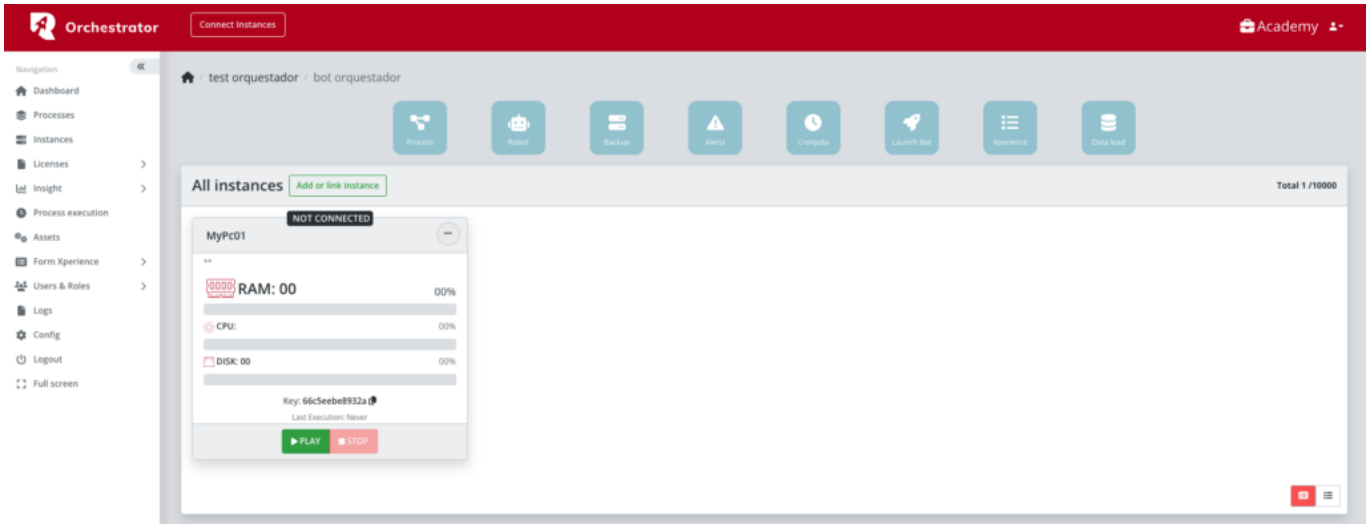
```
event.preventDefault();
const siteKey = ‘nuestroCodigoSiteKey’;
grecaptcha.ready(() => {
grecaptcha
.execute(siteKey, { action: ‘submit’ })
.then(token => {
console.log(‘ Token generado:’, token);
// Inyectar el token en el form
const form = document.getElementsByTagName(‘form’)[0];
let input = form.querySelector(‘input[name=”g-recaptcha-response”]’);
if (!input) {
input = document.createElement(‘input’);
input.type = ‘hidden’;
input.name = ‘g-recaptcha-response’;
form.appendChild(input);
}
input.value = token;
// (Opcional) si tenés una función que controla validaciones / horario:
// enviarFormulario({ target: form, preventDefault: ()=>{} });
// y dentro de enviarFormulario llamás form.submit() al final.
// Disparar el envío final del form
form.submit();
});
});
```



Asi podremos ingresar a nuestro propio recaptcha para que podamos ver la informacion de nuestra cuenta de google de los envios de informacion de nuestros formularios.

Orquestador Rocketbot: Cronjobs Advance

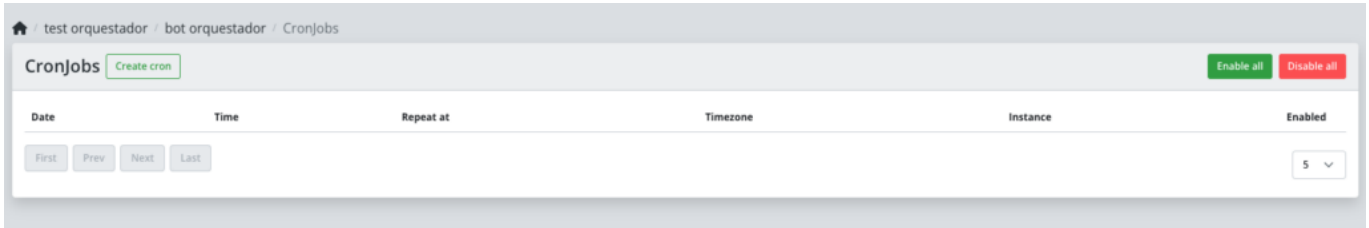
Un **Cronjob** es una tarea programada que permite ejecutar procesos de forma automática en fechas y horas específicas. Una vez que has creado un proyecto y asignado un proceso en el Orquestador, puedes configurar su periodicidad mediante esta herramienta.



Gestión de Cronjobs

Desde la interfaz de Cronjobs, tienes el control total sobre la automatización:

- **Habilitar/Deshabilitar**: Permite activar o pausar todos los cronjobs de forma masiva o individual.
- **Visualización Detallada**: Muestra la información técnica de cada tarea programada, como la instancia de ejecución y la zona horaria.
- **Creación**: Botón dedicado para configurar nuevas tareas programadas.

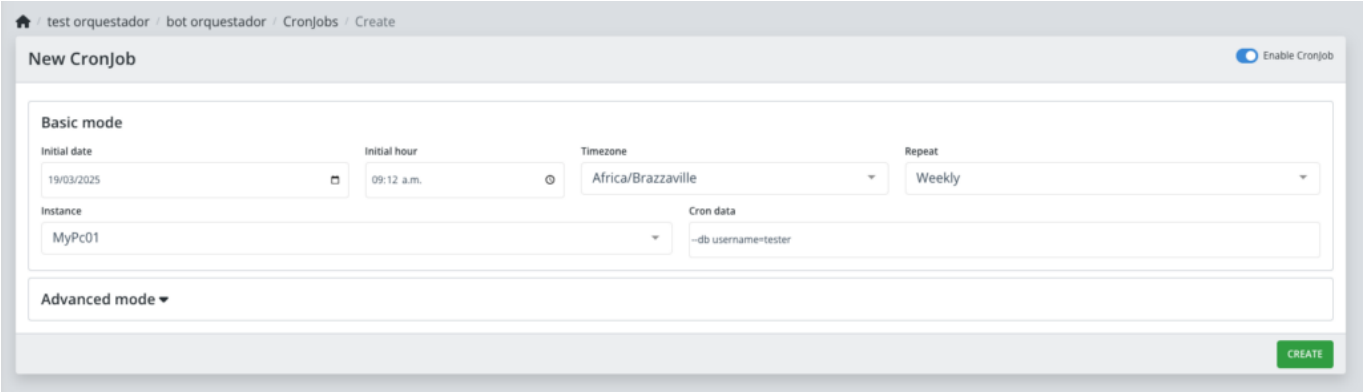


Modos de Configuración

Rocketbot ofrece dos niveles de personalización dependiendo de la complejidad de la tarea:

1. Basic Mode (Modo Básico)

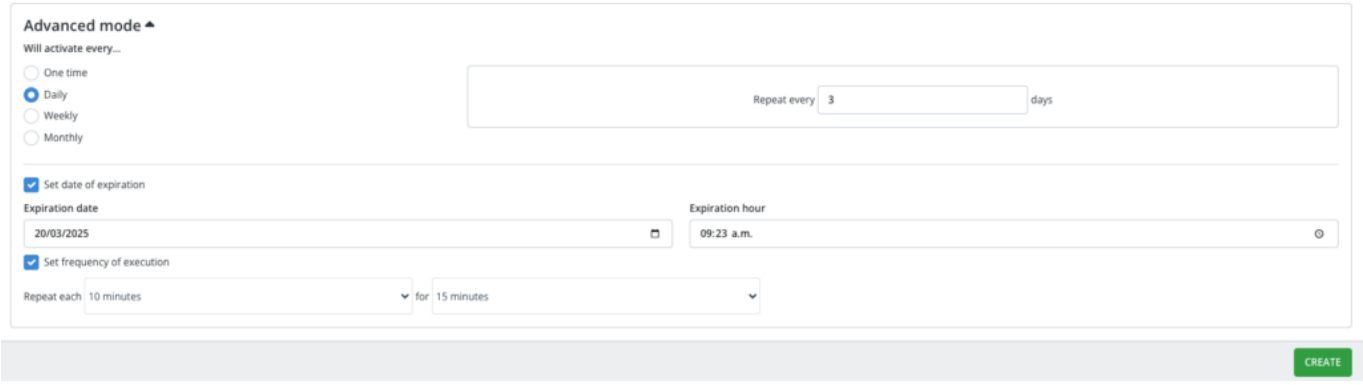
Diseñado para asignaciones rápidas y eficaces donde la periodicidad es estándar. En este modo se configuran los siguientes parámetros:



- **Initial Date**: Define el día, mes y año en que el proceso comenzará a ejecutarse.
- **Initial Hour**: Establece la hora exacta del primer arranque.
- **Timezone**: Parámetro crítico para asegurar que el robot se ejecute en el horario correcto según la región geográfica.
- **Repeat**: Permite elegir frecuencias predefinidas: diario, semanal, mensual, cada 5 o 10 minutos, o ejecución única (no repetir).
- **Instance**: Determina en qué instancia (computadora o servidor) se disparará la tarea.
- **Cron data**: En este apartado pueden ingresarse parámetros adicionales que serán enviados al robot cuando se ejecute a partir de su línea de comandos. Para obtener estos parámetros se puede usar el comando **Command line parameters** del módulo **System** nativo, o en caso de tener una versión anterior de Rocketbot, el módulo SystemAdvanced (**Get arguments**), asignando una variable para capturar la información. Command line también se une a cron data.

2. Advanced Mode (Modo Avanzado)

Este modo proporciona un control granular para escenarios donde la repetición requiere detalles específicos de tiempo.



- **Will activate every…**: Configuración personalizada para activaciones por intervalos de tiempo, días específicos de la semana o fechas mensuales.
- **This cronjob will execute one time on at**: Permite programar una ejecución única detallando con precisión absoluta la fecha y hora.
- **Set date of expiration**: Establece una fecha de “caducidad” para el cronjob, tras la cual dejará de ejecutarse automáticamente.
- **Set frequency of execution**: Control exhaustivo para definir la frecuencia de ejecución minuto a minuto.

En caso de querer mas detalles puede visitar la siguiente Documentación: Trabajar con cronjobs

---

Rocketbot Studio : Conflictos de Librerías

El empleo de diversas librerías en Rocketbot puede dar lugar a incompatibilidades si las versiones requeridas por los módulos personalizados no coinciden con las que ofrece la plataforma. Este informe examina las razones detrás de estos conflictos y sugiere soluciones para solucionarlos.

Importante

Es fundamental considerar la versión de Python utilizada al instalar librerías en Rocketbot. La versión 2020 de Rocketbot opera con Python 3.6.8 de 32 bits, mientras que a partir de la versión 2023 se utiliza Python 3.10 de 64 bits.

En caso de ejecutar módulos mediante un proceso externo, la librería correspondiente debe estar instalada en el entorno desde el cual se realiza la ejecución.
Esto permite operar dentro del entorno adecuado, evitar incompatibilidades y asegurar la correcta recepción de la información en Rocketbot.

Causas de los Conflictos

Los conflictos pueden surgir por diversas razones, como la existencia de diferentes versiones de librerías que no coinciden con las de Rocketbot. También pueden presentarse problemas de compatibilidad entre dependencias, ya que múltiples versiones de una misma librería pueden causar errores. Además, la instalación de versiones adicionales puede interferir con las librerías predeterminadas de Rocketbot.

Soluciones Recomendadas

Renombrar el Archivo de la Librería:

En casos donde se vea errores como el siguiente:

```
cannot import name 'deprecated' from 'typing_extensions' (C:\Rocketbot_win_20240528\typing_extensions.pyc)
```

typing_extensions es una librería que viene compilada con Rocketbot y este error suele darse porque viene con una versión más antigua que la instalada para el script, por lo que al ejecutar otra librería que tiene como dependencia typing_extensions, rompe.

En estos casos se recomienda indicar la ruta hacia la lib que instalaron o si el error persiste como workaround se puede renombrar la carpeta de la librería importada o el nombre del archivo.py por uno diferente, por ejemplo en lugar de llamarlo **typing_extensions.py** lo llamaremos **new_typing_extensions.py**. En tu código deberás encontrar los imports que hagan tu scripts o las dependencias que importas cambiando lo siguiente:

import typing_extensions por **import new_typing_extensions as typing_extensions** (así se mantiene el nombre original)

Y también (teniendo en cuenta el error del ejemplo):

from typing_extensions import deprecated por **from new_typing_extensions import deprecated**

Con estos cambios, el script ejecutado por Rocketbot entenderá que no debe utilizar typing_extensions que viene nativamente con Rocketbot, sino **new_typing_extensions** instalado manualmente por nosotros.

Este ejemplo se puede aplicar tanto a archivos .py como a carpetas de librerías. Por ejemplo si queremos utilizar una nueva versión de la librería pandas, podemos descargarla localmente, renombrar la carpeta a new_pandas y luego en nuestro script la importamos con ese nombre.

Uso de Procesos Externos:

En los casos donde la solución anterior no funcione, se sugiere ejecutar el proceso mediante popen o a través de un webhook. Esto permite el uso de la versión de Python y librerías del ambiente donde se ejecute en vez de usar los de Rocketbot.

Popen

Para ejecutar su script mediante un Popen realice lo siguiente en un comando Ejecutar Python:

```
from subprocess import Popen, PIPE
process = subprocess.Popen(
        ['python', 'ruta/a/su/script.py', 'arg1', 'arg2', 'argN'],
```

```
        stdout=subprocess.PIPE,
        stderr=subprocess.PIPE
    )
out, err = process.communicate()
print(out.decode())
```

En su script imprima en consola para obtener lo que necesita en Rocketbot. En el ejemplo se ejecuta un script donde se le envían argumentos (que bien podrían ser datos de variables de Rocketbot). Luego en el script que está ejecutando toma los argumentos mediante: **sys.argv**
Esto trae cada valor de la lista que se le pasa al popen: **sys.argv[0]** traería python, **sys.argv[1]** ruta_script.py, **sys.argv[2]** arg1 y así según se necesite. Es opcional para el caso en que se necesite enviar datos de Rocketbot al script.

Webhook

Al levantar un puerto local se queda en pausa a la escucha de algún evento POST o GET.

- Es un módulo que se descarga del market, el manual se encuentra en la carpeta example del módulo, en la ruta Rocketobot/modules/Webhook
- Puedes levantar un endpoint propio donde enviar información y que el robot escuche o usar ngrok para levantar uno (pasos en el manual).
- Al ejecutar el comando de 'Crear webhook', Rocketbot se pausará esperando que se consulte la url configurada en el comando, ya sea con una petición GET o POST.

En el contexto de conflicto de librerías, un workaround utilizando Webhook es crear un .bat de la siguiente manera:

C:/Python312/python.exe "C:\Users\user\Desktop\archivo.py"

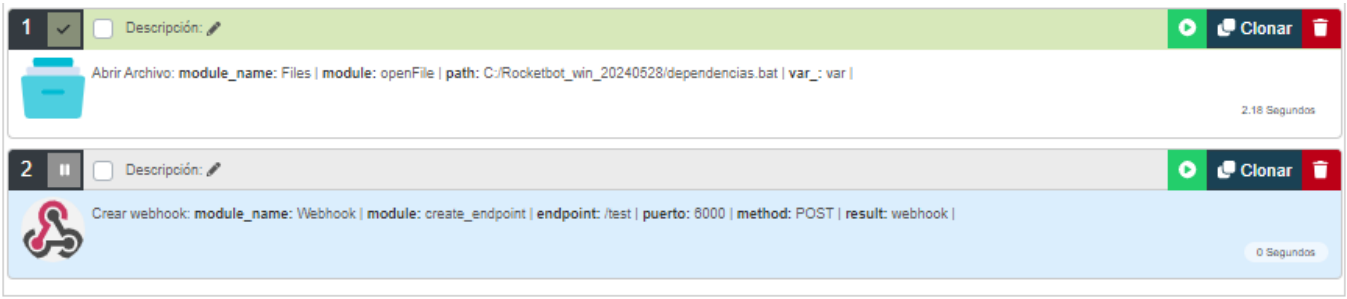
Cambiando el path que está entre comillas por el path hacia el script a ejecutar y el primer path por el path hacia el ejecutable de Python en el ambiente a utilizar.

Dentro del archivo.py que contiene el script a ejecutar se tiene que agregar al principio las importaciones de request y un sleep que dará tiempo a ejecutar el comando de webhook. Al final del script se tiene que agregar la petición post para recibir la información mediante webhook al endpoint levantado:

```
import requests
from time import sleep
sleep(10)
(acá va el código)
requests.post("http://127.0.0.1:6000/test", data={"var1":"data1", "var2":
"data2"})
```

Cambiando 127.0.0.1:6000 por el host y puerto elegido. El parámetro data en este ejemplo es opcional, se puede utilizar si se requiere enviar información a Rocketbot.

En el bot se tienen que crear los siguientes comandos uno seguido del otro:



El comando de webhook se tiene que ejecutar dentro de los segundos declarados en el sleep del archivo.py, en este ejemplo se tiene que ejecutar antes de los 10 segundos.

Ejecución de Automatizaciones desde Línea de Comandos y API

Este documento proporciona una guía rápida para ejecutar procesos y bots en las plataformas de automatización más populares: **UiPath**, **Automation Anywhere** y **Power Automate** desde **Orquestador de Rocketbot.**

Debes crear un paquete para ejecutar desde el orquestador como se indica aqui Subir archivos con extensión .zip al robot puedes utilizar la siguiente guia para agregar comandos que llamen a estas herramientas de RPA

UiPath

Ejecución desde Línea de Comandos

UiPath proporciona una herramienta llamada UiRobot.exe para ejecutar procesos sin necesidad de Orchestrator.

Comandos básicos:

```
UiRobot.exe execute --process <NombreDelProceso>
UiRobot.exe execute --file "C:\Ruta\Al\Archivo.xaml"
UiRobot.exe execute --process <NombreDelProceso> --input
"{'param1':'valor1','param2':2}"
UiRobot.exe execute --process <NombreDelProceso> --folder CarpetaOrchestrator
```

- --file: Ejecuta directamente un .xaml local.
- --input: Parámetros en formato JSON.
- --folder: Carpeta desde Orchestrator.

 Documentación CLI UiPath

☁ Ejecución mediante API (UiPath Orchestrator)

UiPath Orchestrator permite controlar procesos de forma remota mediante API REST.

1. Autenticación:

```
POST /api/account/authenticate
{
  "tenancyName": "default",
  "usernameOrEmailAddress": "user",
  "password": "pass"
}

Respuesta: {"result": "token"}
```

2. Obtener Release Key:

```
GET /odata/Releases
Authorization: Bearer <token>
```

3. Ejecutar el proceso:

```
POST /odata/Jobs/UiPath.Server.Configuration.OData.StartJobs
Authorization: Bearer <token>
{
  "startInfo": {
    "ReleaseKey": "<releaseKey>",
    "Strategy": "Specific",
    "RobotIds": [1234],
    "InputArguments": "{"param":"value"}"
  }
}
```

 Documentación API UiPath

🤖 Automation Anywhere

🖥️ Ejecución desde Línea de Comandos

Utiliza la herramienta aascli:

```
aascli run --bot "Bots/MyBotFolder/MyBot" --params '{"param1":"valor1"}'
```

- El bot debe estar publicado en Control Room.
- Necesitas estar autenticado con Bot Agent activo.

🔗 CLI de Automation Anywhere

🌐 Ejecución mediante API

1. Autenticación:

```
POST /v1/authentication
{
  "username": "user",
  "password": "pass"
}
```

Respuesta: {"token": "JWT_TOKEN"}

2. Obtener IDs necesarios:

```
GET /v3/automations
GET /v3/users
```

3. Ejecutar el bot:

```
POST /v3/automations/deploy
Authorization: Bearer <token>
{
  "fileId": "<bot_id>",
  "runAsUserIds": ["<user_id>"],
  "botInput": {
    "param1": "value1"
  }
}
```

🔗 API Automation Anywhere

☁️ Power Automate

Flujos de Escritorio desde Línea de Comandos

Usa PAD.Console.Host.exe:

```
"C:\Program Files (x86)\Power Automate Desktop\PAD.Console.Host.exe" "ms-powerautomate:/console/flow/run?workflowName=NombreDelFlujo"
```

También puedes ejecutar por ID:

```
PAD.Console.Host.exe "ms-powerautomate:/console/flow/run?workflowId=<id>&inputArguments={\"param1\":\"value1\"}&autologin=true"
```

Documentación CLI PAD

Ejecución mediante API

1. Autenticación con Azure AD:

Utiliza Microsoft Identity Platform para obtener un token con permisos para acceder a Dataverse.

2. Obtener Workflow ID:

```
GET https://<org>.crm.dynamics.com/api/data/v9.2/workflows?$filter=category eq 6
```

3. Ejecutar flujo:

```
POST https://<org>.crm.dynamics.com/api/data/v9.2/workflows(<workflowid>)/Microsoft.Dynamics.CRM.RunDesktopFlow
Authorization: Bearer <token>
{
  "runMode": "attended",
  "runPriority": "normal",
  "connectionName": "Default Windows",
  "timeout": 7200,
  "inputs": "{\"param1\":\"value1\"}"
}
```

API Power Automate

Recursos Recomendados

- UiPath sin Orchestrator (YouTube)
- Automation Anywhere desde API (YouTube)
- Power Automate ejecución directa (YouTube)

Rocketbot Studio : Migrar bots con Virtualizacion

Pasos a Seguir para la Migración de Bots de Virtualización

Virtualización en la Máquina Destino

Es recomendable realizar la virtualización directamente en la máquina o computadora final. Esto se debe a que la resolución de pantalla puede variar, lo que podría requerir ajustes en la configuración, similar a la máquina donde se realizó la virtualización inicialmente.

Consistencia de Imágenes de Referencia

Es posible que la imagen utilizada como referencia no sea la misma que la que se encuentra en la computadora de destino. En tal caso, será necesario capturar nuevamente las imágenes de referencia, incluyendo focos , como ejemplo, el comando “hacer clic en imagen” , etc.

Compatibilidad de Resolución y Zoom

Si la máquina en la que se trabaja es compatible tanto con la resolución como con el zoom utilizados al momento de capturar las pantallas en la referencia de virtualización, se deben seguir las siguientes consignas para asegurar una migración exitosa de los bots:

- **Registro de Comandos**
  Al grabar un comando en el entorno de virtualización, las imágenes se almacenarán en una carpeta específica dentro del directorio de Rocketbot, bajo el nombre correspondiente al bot, en la ruta Rocketbot/robots. Esta carpeta es crucial, ya que las imágenes se utilizan como comandos en Rocketbot Studio, tales como “hacer clic en imagen” y “esperar por imagen”.

- **Migración del Archivo .db**
  Para completar la migración del bot, es imprescindible transferir también el archivo .db. Este archivo es un componente básico del proceso de migración.

Consideraciones Finales

Asegurarse de seguir meticulosamente los pasos anteriores es esencial para lograr una migración exitosa de los bots de virtualización.

Una correcta transferencia de los componentes no solo garantizará que el bot funcione adecuadamente en su nuevo entorno, sino que también optimizará su rendimiento y facilitará futuras actualizaciones.

Adoptar un enfoque riguroso en este proceso es clave para minimizar cualquier inconveniente y asegurar un despliegue eficiente.

En caso de no entender algún concepto relacionado con la virtualización en Rocketbot Studio, se recomienda consultar la documentación : Documentación de Virtualización en Rocketbot.

Documentación Robot Template

Rocketbot necesita facilitar la revisión y aprobación en el menor tiempo posible de los BOTs enviados por los Partners para su publicación en el Marketplace. Se hace necesario estandarizar el proceso y la forma de construcción, por esta razón se ha habilitado el nuevo módulo ProbeForms en Rocketbot Studio, que les permitirá resolver la interfaz hacia el usuario y el ingreso de los datos requeridos para la operación del BOT. La idea es que los construyan usando este modulo de ingreso de datos y no desarrollar su propia solución, esto para agilizar la revisión y aprobación del mismo. Como una guía para ustedes se construyó este BOT con el fin de usarlo como plantilla para la creación de otros robots en MarketBot, proporcionando una estructura organizada para desarrollar robots especializados. Facilita la creación enfocados en dividir tareas específicas, como configuración, trabajo con Excel, envío de correos, etc. además de validación e ingreso de información a través de un formulario local, optimizando así el uso por parte del usuario. El proceso de ejemplo de este robot recorre y obtiene la información de un Excel para luego llenar un formulario web, obteniendo la respuesta y volviendo al Excel a indicar si fue procesado o no, finalmente envía un correo con el archivo adjunto.

INPUTS

VARIABLES MODIFICABLES

Robot	Variable	Valor Actual	Explicación
main	{espera_datos}	120	Son los 120 segundos de espera a que el usuario ingrese la información en el formulario, esta variable es utilizada en los comandos **Wait submit** del módulo **ProbeForms**

ESTRUCTURA

```
Proyecto —
|- docs (no usada)
|- logs
|- logs_bot
|- modules (no usada)
|- resources
|- robots (no usada)
|- screenshot (no usada)
|- main.bat
|- robot.db
|- README.md
|- package.json
```

MÓDULOS

Nombre	Versión
SystemAdvanced	2.8
ProbeForms	1.2.0
gmail_	1.12.7

ROBOTS

main

Robot principal, encargado de dejar todas las variables limpias antes de iniciar, posteriormente llama a los robots hijos para ejecutar cada parte del proceso.

config

Llama al robot para validar los procesos y valida si faltan datos para la configuración, si faltan o es la primera ejecución (todas las variables de config vacías) llamará al robot encargado del ingreso, sino, mostrará el formulario de reinicio para que el usuario decida si quiere reiniciar la configuración e ingresar de nuevo los datos o si prefiere continuar con los datos que ya se tienen.

termina_proceso

Valida si existen procesos de ProbeForms activos y los mata si corresponde.

ingresa_config

Muestra el formulario de ingreso de información y obtiene los datos ingresados por el usuario.

lee_excel

Abre el archivo Excel y extrae la información.

recorre_datos

Recorre los datos obtenidos del Excel y los guarda en sus variables correspondientes, dentro del ciclo llama al robot encargado de ir llenando el formulario web con estos datos, finalmente cierra el navegador y guarda los cambios en el Excel.

llena_form

Llena la información extraída del archivo Excel en el formulario web, valida la respuesta entregada y escribe el status en la celda correspondiente.

envia_correo

Envía el correo a el/los destinatarios, adjuntando el archivo Excel.

escribe_log

Escribe la información enviada desde los robots en el archivo log customizado, además valida la información de la variable {detener}, si es True dentendrá todo el proceso.

valida_res

Valida la respuesta de la variable {res}, si es False, cerrará el navegador y llamará al robot para escribir el log.

VIDEO EJECUCIÓN

¿CÓMO PROBARLO?

1- Descarga el siguiente .zip □□ y descomprimelo en la carpeta que desees

template_marketbotDownload

2- Descomprime el siguiente .zip □□ que contiene los módulos a utilizar

mods-templateDownload

3- Las 3 carpetas de los módulos (gmail_, ProbeForms y SystemAdvanced) pégalos en tu_carpeta_rocketbot/modules

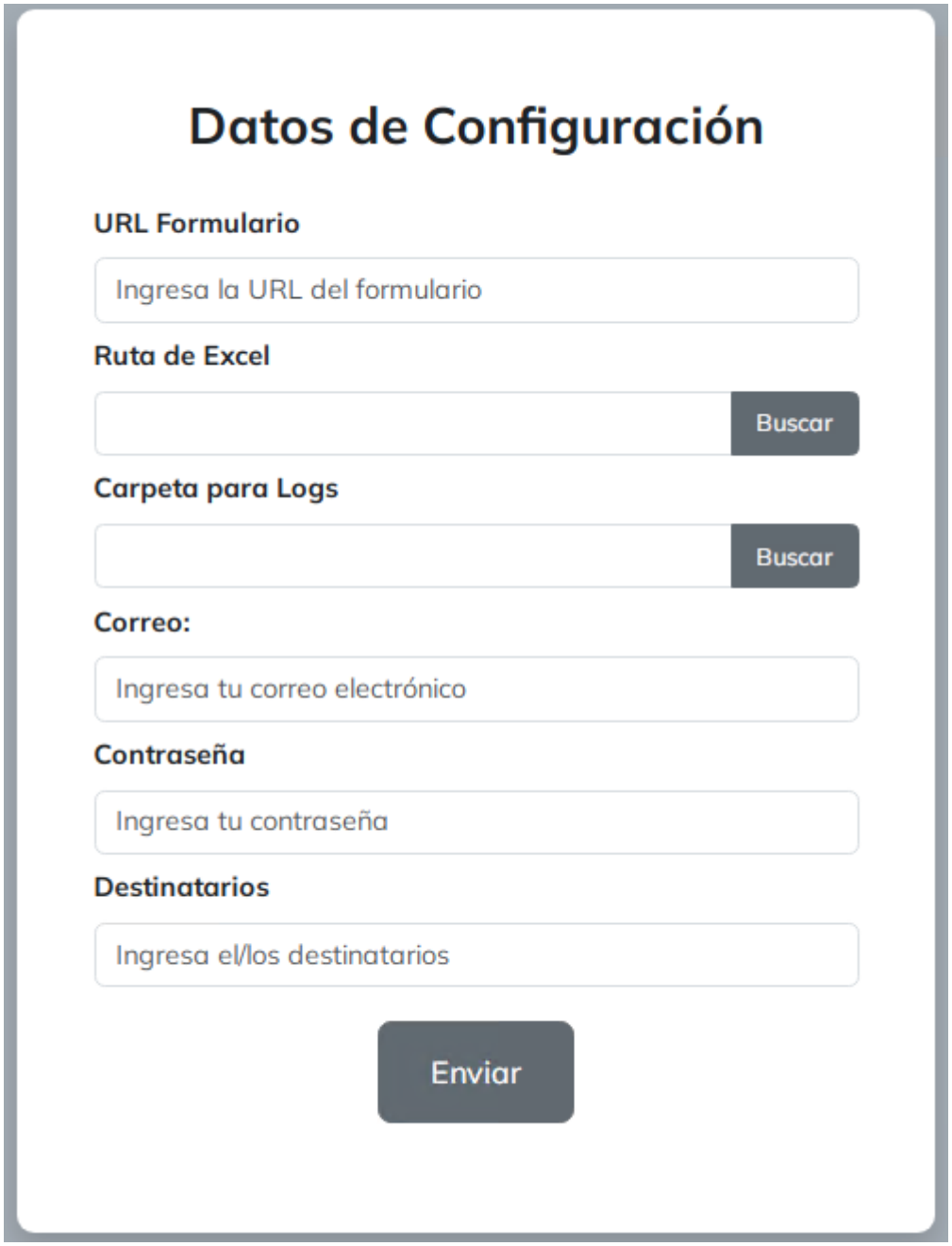
4- Abre Rocketbot Studio y carga la base de datos robot.db que está en la raíz de la carpeta template_marketbot

5- Se mostrará una lista de robots, clickea en el que se llama main

6- Ejecuta el robot con el botón verde → Ejecutar

7- Te mostrará el formulario inicial donde deberás ingresar tu información

□ Considerar que los datos se deben ingresar en un tiempo máximo de 2 minutos, si no se ingresan dentro de ese rango, verás una ventana de que el robot se detuvo y esto significa que finalizó con error. Este tiempo está configurado por defecto, si necesitas aumentarlo, debes modificar la variable {espera_datos} en el robot main



URL Formulario: Debe ser https://roc.myrb.io/dev/forms/INGKKJ0HGGXOJNRZ
Ruta de Excel: debes buscar la ruta de tu carpeta template_marketbot/resources y elegir el archivo datos_form.xlsx
Carpeta para Logs: debes buscar la ruta de tu carpeta template_marketbot/logs_bot
Correo: Debes ingresar el correo desde donde se enviará la información, debe ser un **gmail**
Contraseña: Debes ingresar la ***contraseña de aplicación*** de la cuenta gmail
Destinatarios: Debes ingresar el correo del destinatario, si es más de uno los debes separar por coma

8- Una vez ingresada el robot continuará con la ejecución

9- Una vez finalizada la ejecución clickear el botón Guardar, realizar esto sólo en la primera ejecución.

 Con esto en la próxima ejecución preguntará si se quieren reiniciar o no

los datos de configuración, donde si se elige la opción SI, mostrará un formulario para volver a ingresarlos, en caso contrario el robot continuará con los datos previamente ingresados, ésto para evitar estar escribiendo lo mismo en cada ejecución y sólo realizarlo en caso de que alguna información cambie.