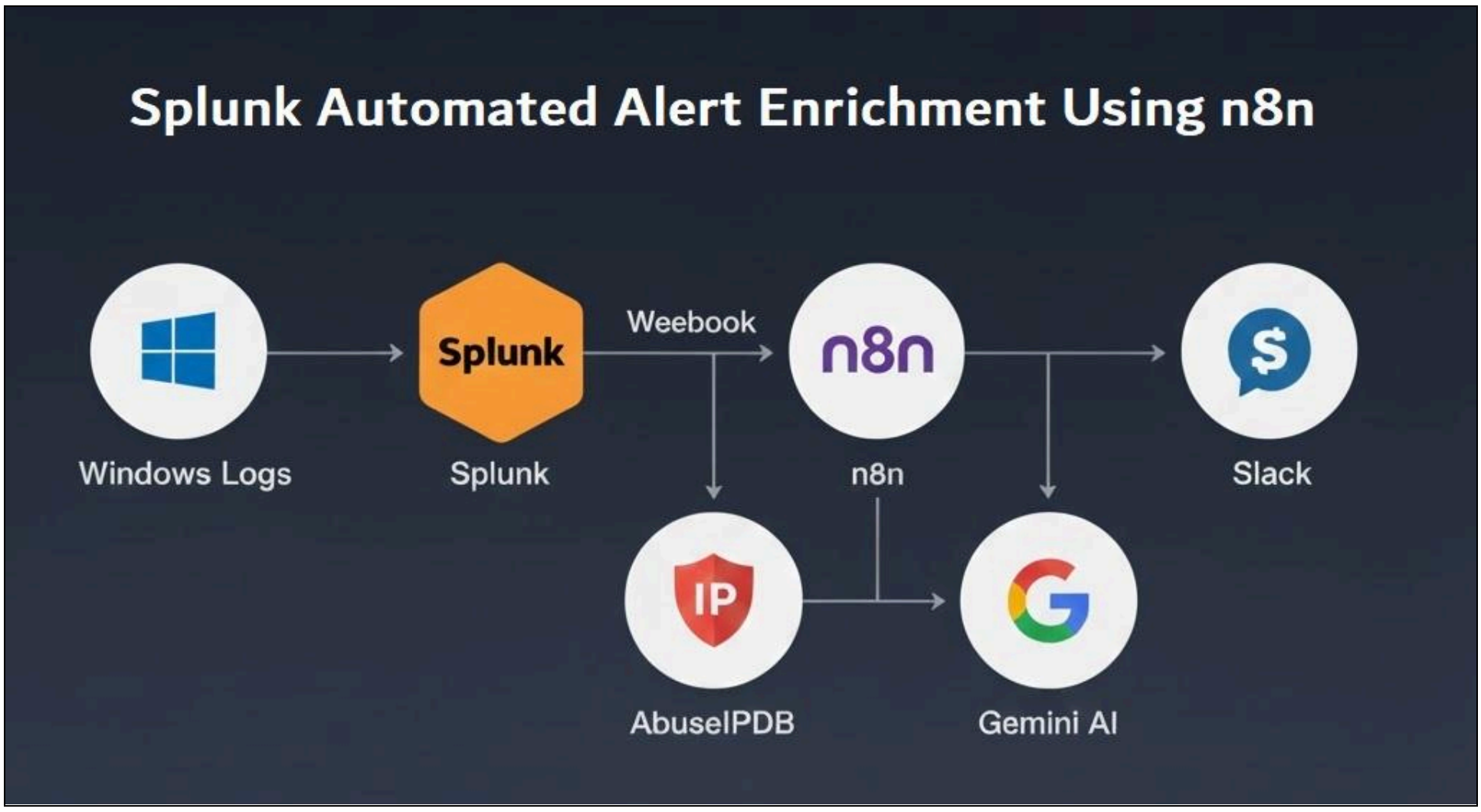
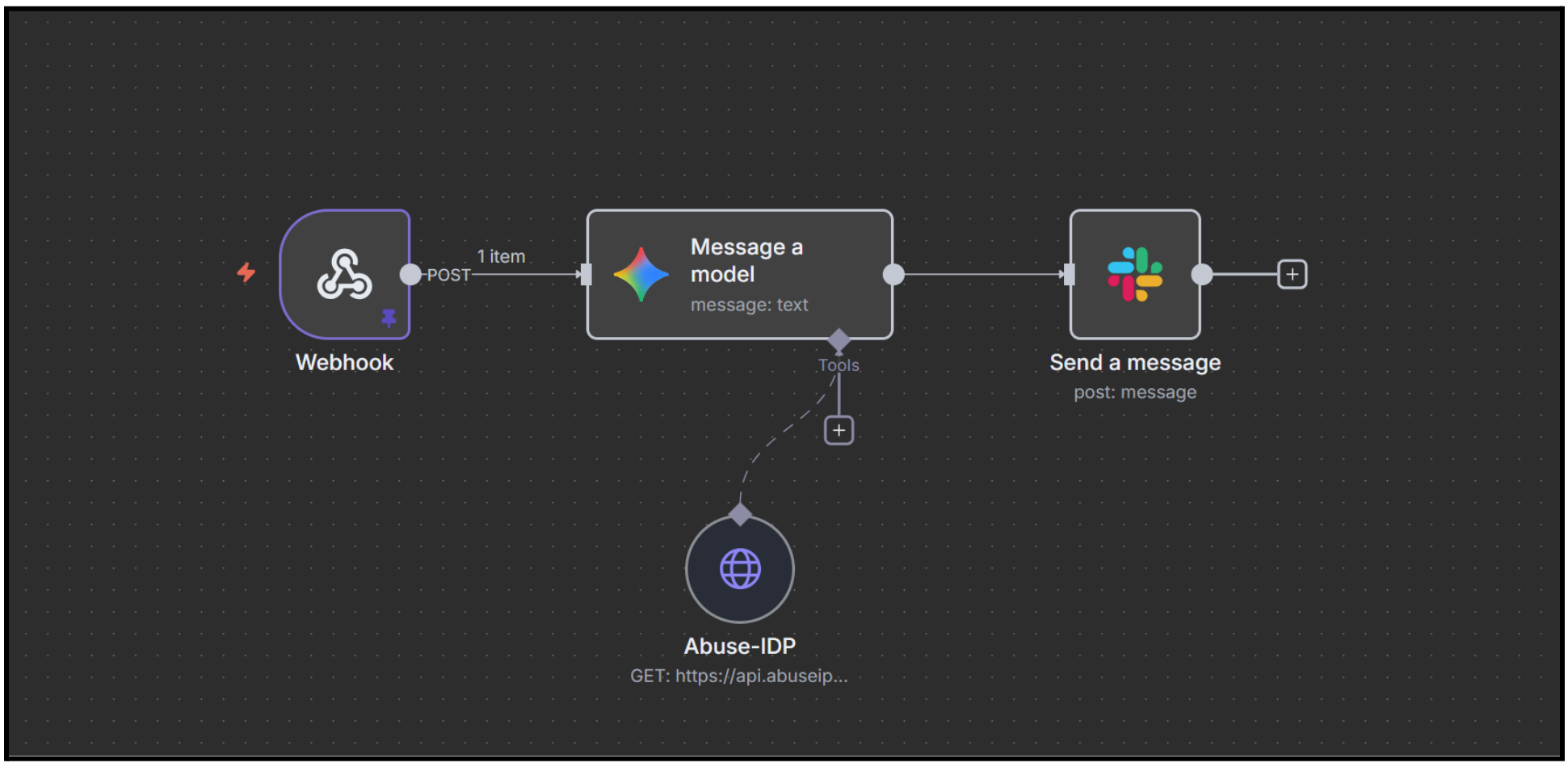


Splunk Alert Enrichment Workflow with n8n and AbuseIPDB

Overview

This project builds an automated alert enrichment pipeline using Splunk, n8n, AbuseIPDB, Slack, and Gemini AI. Windows logs are collected through the Universal Forwarder, ingested into Splunk, forwarded to n8n through webhooks, enriched with threat intelligence and AI analysis, and then delivered to Slack for quick triage.



Project Objectives

By completing this project, you will learn how to build an end-to-end automated alert enrichment workflow using Splunk, n8n, AbuseIPDB, and Slack. You will be able to:

- Deploy a Windows 11 virtual machine and Ubuntu Server virtual machines for Splunk Enterprise and n8n
- Monitor Windows endpoints using Splunk and configure custom indexes for log storage
- Install and manage the Splunk Universal Forwarder for reliable log collection
- Detect Windows authentication events, including failed logons such as Event ID 4625
- Use n8n to receive Splunk alerts through webhooks and automate enrichment steps
- Perform IP reputation checks by integrating the AbuseIPDB API
- Forward enriched and structured security alerts to Slack for quick analyst triage

Prerequisites / Requirements

Technical Requirements

Before starting, ensure that **VMware Workstation** is installed. It's now free to use and will act as the virtualization platform for this project. You will need the following components deployed and accessible within your lab environment:

- A Windows 11 virtual machine with:
    - Splunk Universal Forwarder installed
    - Sysmon (with a suitable configuration file) installed
- A Splunk Enterprise instance running on Ubuntu Server
- An n8n instance running on Ubuntu Server
- A Slack workspace with an OAuth token for API access
- An AbuseIPDB account with a valid API key
- Access to the Gemini API for AI based enrichment
- Basic network connectivity between all systems to allow log forwarding and webhook communication

Skills Requirements

Before starting, ensure you are familiar with the following:

- Configuring Splunk inputs, indexes, and webhook alert actions
- Creating and managing workflows in n8n
- Understanding how threat intelligence lookups work, including IP reputation scoring
- Interpreting Windows event logs, especially authentication and security events

Setting Up the Windows 11 Virtual Machine

In this section, you will deploy a Windows 11 virtual machine that will act as the monitored endpoint in your security automation workflow. This VM will forward logs to Splunk using the Universal Forwarder and Sysmon.

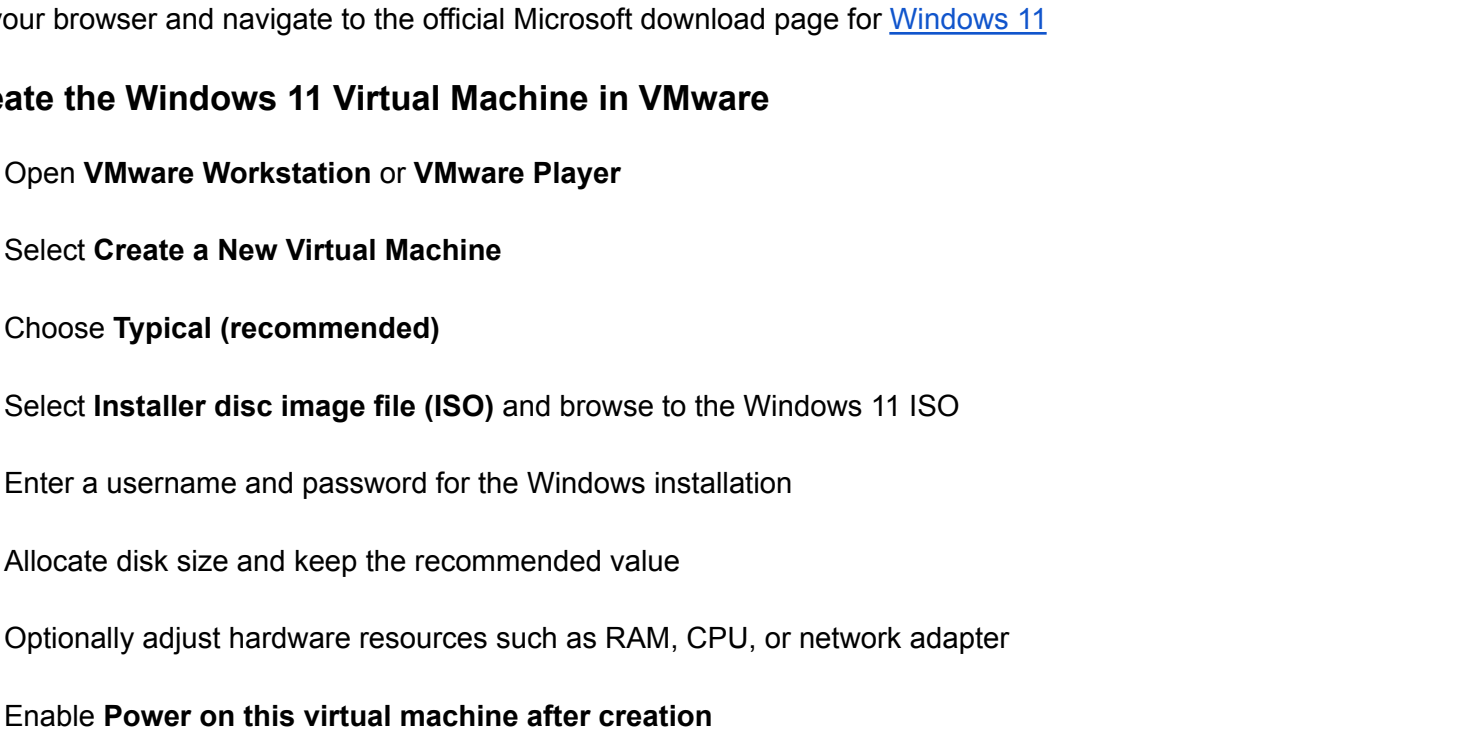
Open your browser and navigate to the official Microsoft download page for Windows 11

1. Create the Windows 11 Virtual Machine in VMware

1. Open **VMware Workstation** or **VMware Player**
2. Select **Create a New Virtual Machine**
3. Choose **Typical (recommended)**
4. Select **Installer disc image file (ISO)** and browse to the Windows 11 ISO
5. Enter a username and password for the Windows installation
6. Allocate disk size and keep the recommended value
7. Optionally adjust hardware resources such as RAM, CPU, or network adapter
8. Enable **Power on this virtual machine after creation**
9. Click **Finish**

2. Complete the Windows 11 Installation

1. The VM will boot from the ISO
2. Follow the on screen installation steps
3. Log in using the credentials you created
4. Install **VMware Tools** if prompted for improved performance and networking
5. Open Command Prompt on the Windows 11 VM and run `ipconfig` to find its IP address.



At this stage, you have a fully functional Windows 11 endpoint ready for log forwarding and event monitoring.

Installing Sysmon on the Windows 11 VM

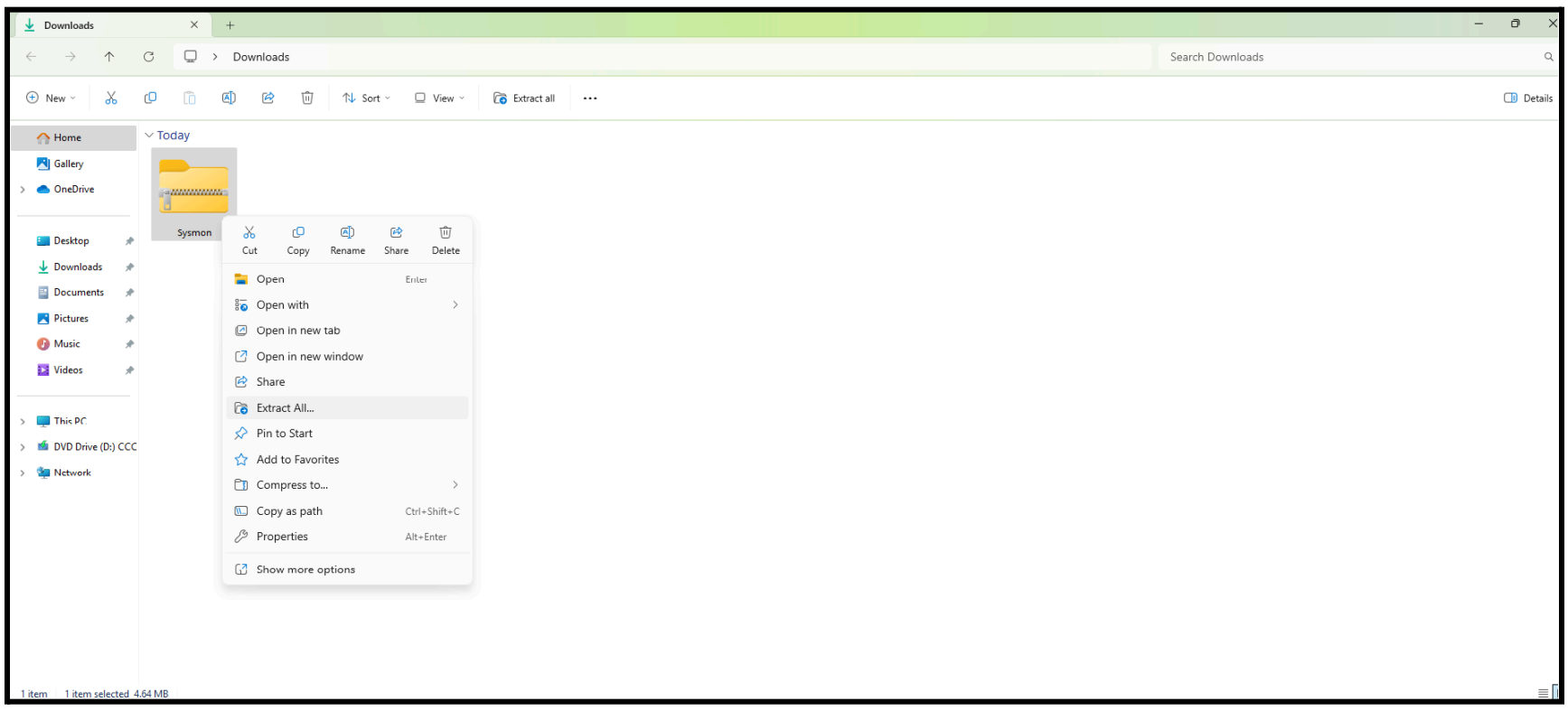
Sysmon is required for generating detailed host level telemetry such as process creation, network connections, and file modification events. These logs will later be forwarded to Splunk.

Why Sysmon?

Sysmon extends native Windows logging by capturing high fidelity security events that are essential for threat detection and enrichment. In this project, Sysmon provides the endpoint telemetry used for automated alert processing.

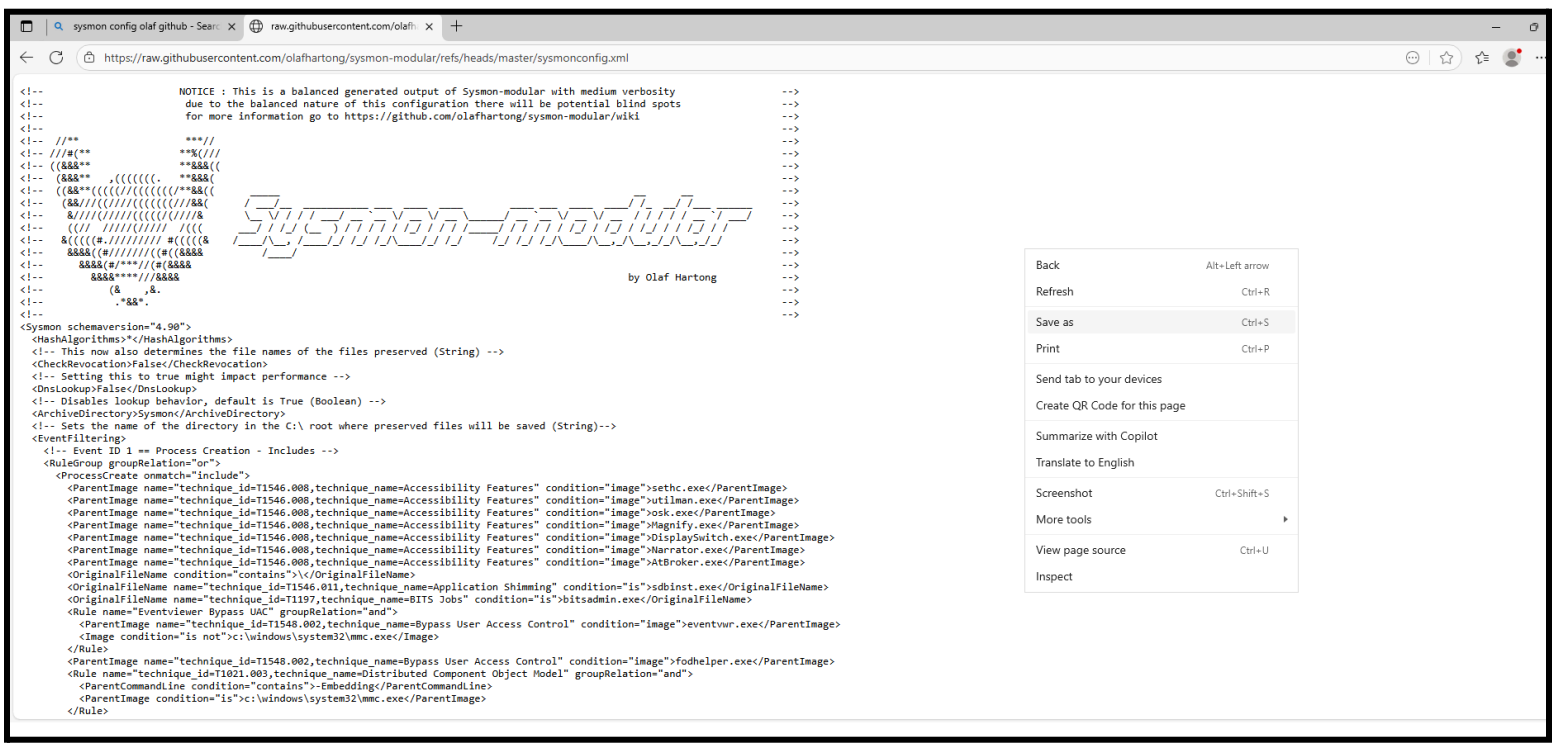
1. Download Sysmon

1. Inside the Windows VM, open Microsoft Edge
2. Navigate to the official Sysinternals Sysmon download page
3. Download the Sysmon package
4. Extract the contents of the ZIP file into a folder



2. Download a Sysmon Configuration File

1. Open your browser and go to the **Sysmon Modular** configuration repository by Olaf Hartong on GitHub
2. Right click on the XML configuration file and select **Save As**



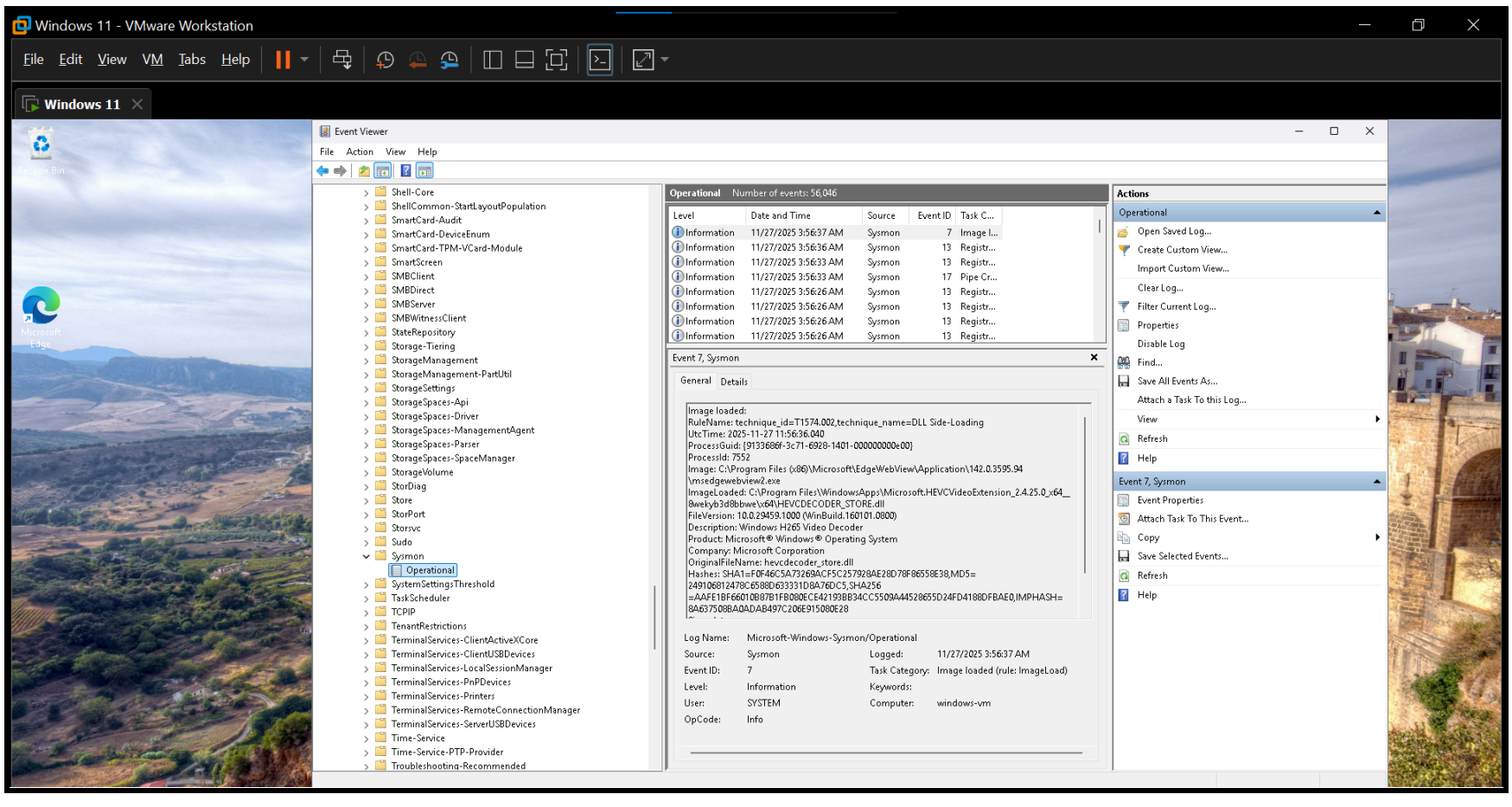
3. Save the file as **sysmonconfig.xml**
4. Move the saved file into the same directory as the Sysmon executable

3. Install Sysmon with the Configuration

1. Open **PowerShell** as Administrator
2. Navigate to the folder containing Sysmon
3. Run the installation command: `.\Sysmon64.exe -i sysmonconfig.xml`
4. Accept the license agreement
5. Verify Sysmon is running: `Get-Service -Name sysmon64`
6. If the service status is **Running**, the installation is successful

To confirm log generation, open **Event Viewer** and navigate to:

Applications and Services Logs > Microsoft > Windows > Sysmon >Operational

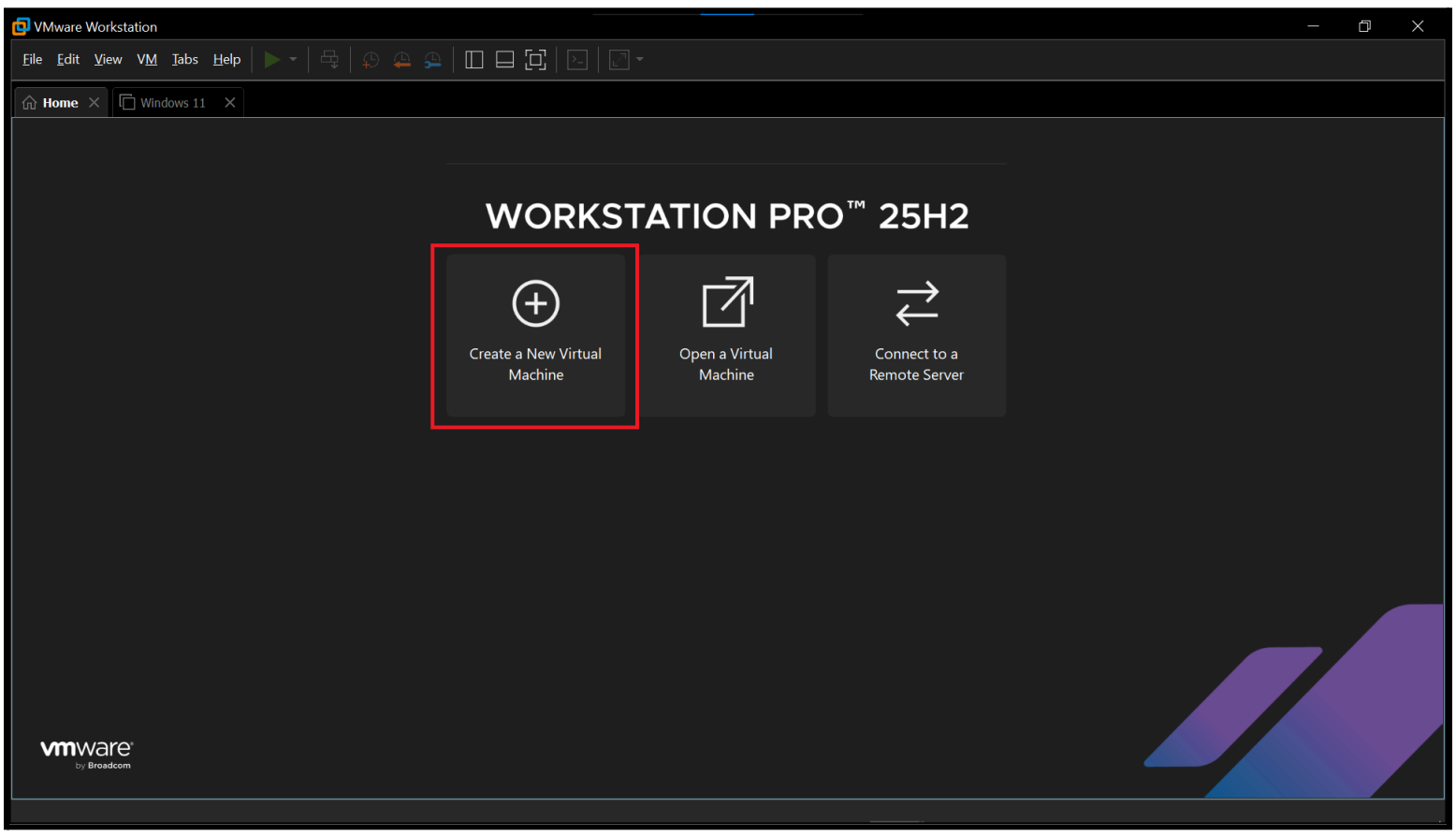


Setting Up the Ubuntu Server Virtual Machine for Splunk

In this section, you will create an Ubuntu Server virtual machine that will host your Splunk Enterprise instance.Download the latest Ubuntu Server ISO from the official Ubuntu website.

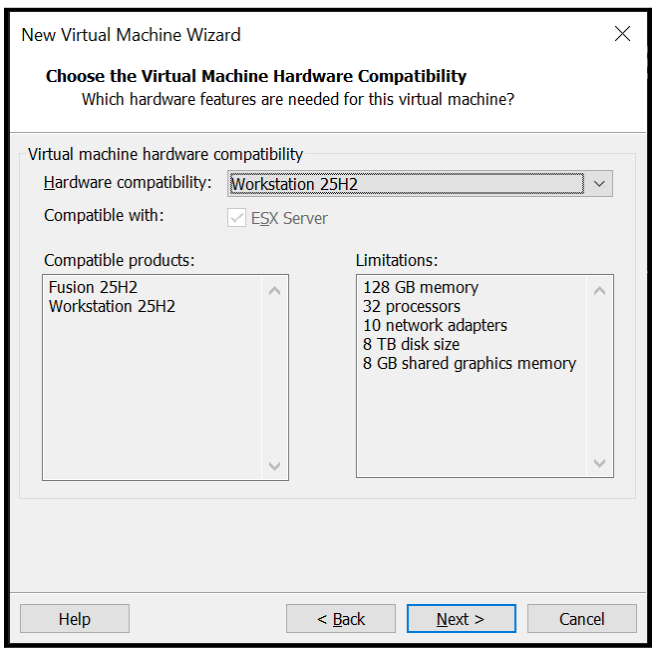
1. Create a New Virtual Machine in VMware Workstation

1. Open VMware Workstation.
2. Select **Home** from the left panel.
3. Click **Create a New Virtual Machine**.

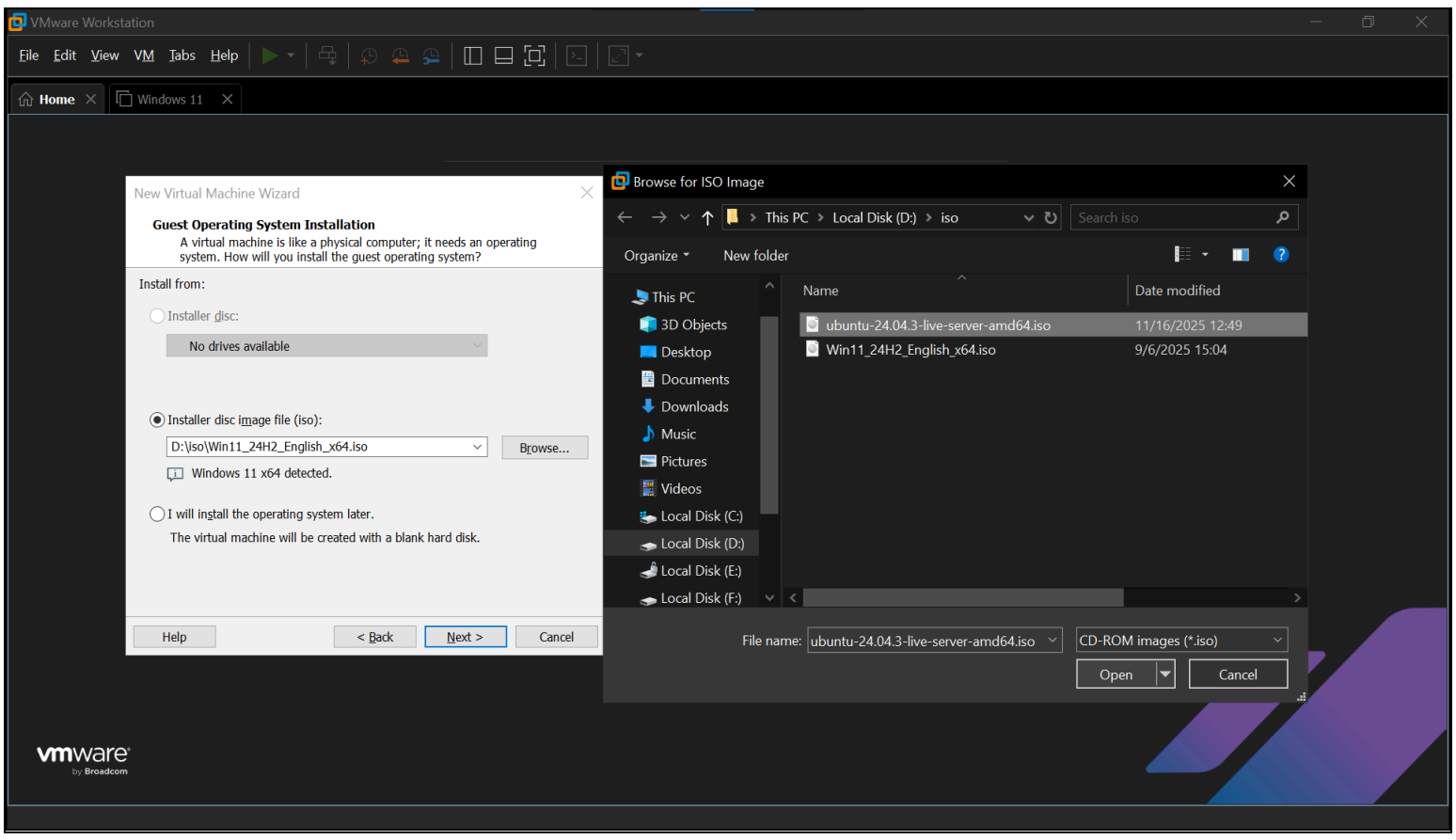


4. Choose **Custom (Advanced)** and select **Next**.

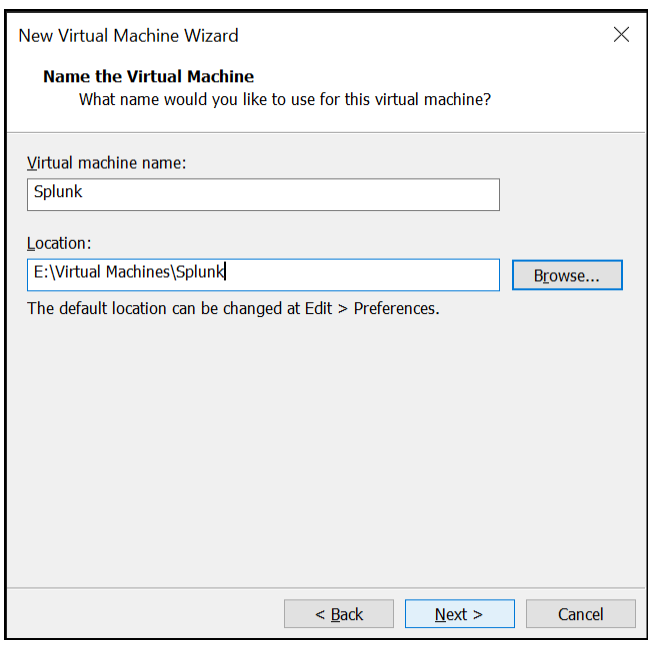
5. Leave the hardware compatibility as default and select **Next**.



6. Select **Installer disc image (ISO)**, then browse to and select the Ubuntu Server ISO you downloaded.Select **Next**.

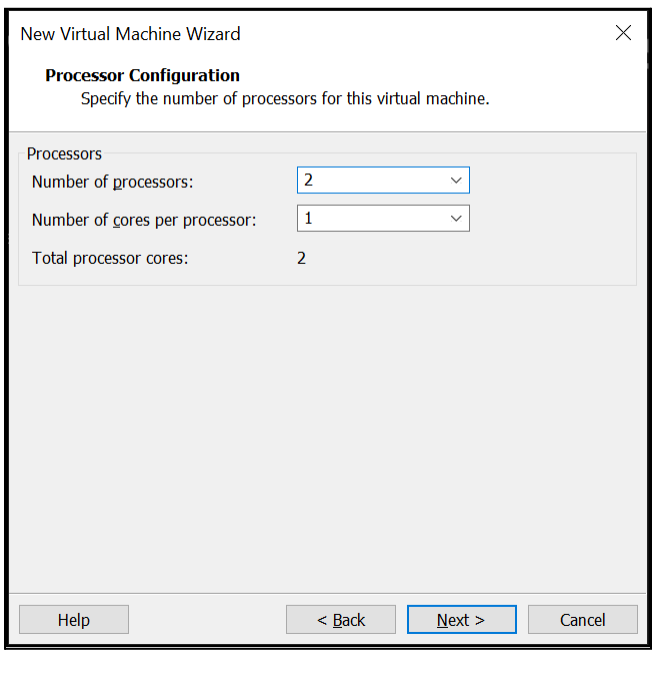


7. Name the virtual machine, choose the location where the VM files will be stored,select **Next**.

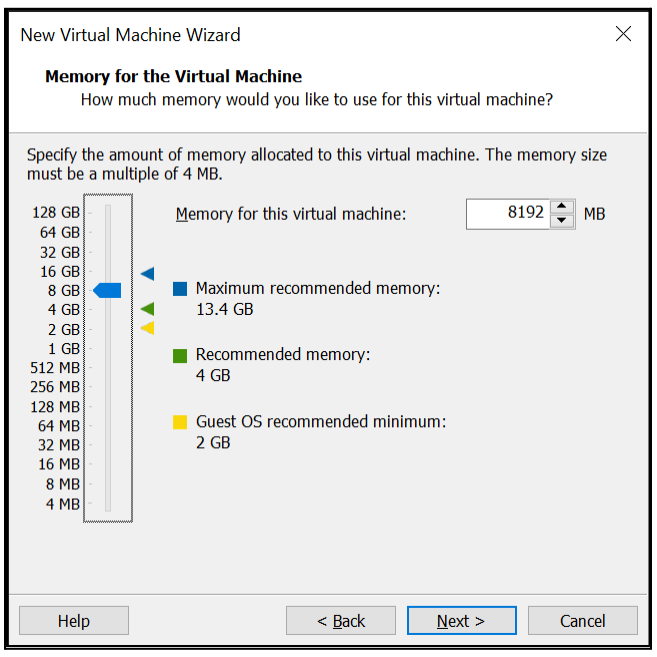


2. Configure CPU and Memory

1. Set the number of processors to **2**.Set the number of cores per processor to **1** and select **Next.**

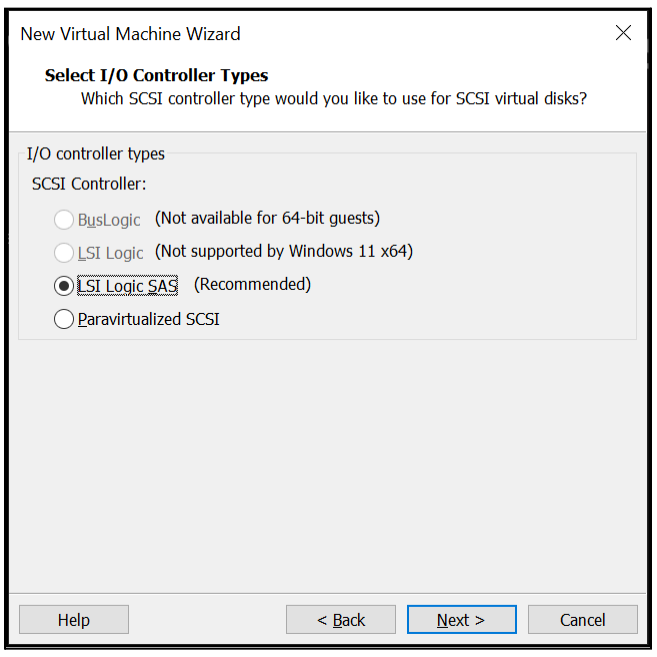


2. Assign **8 GB** of memory and select **Next**.

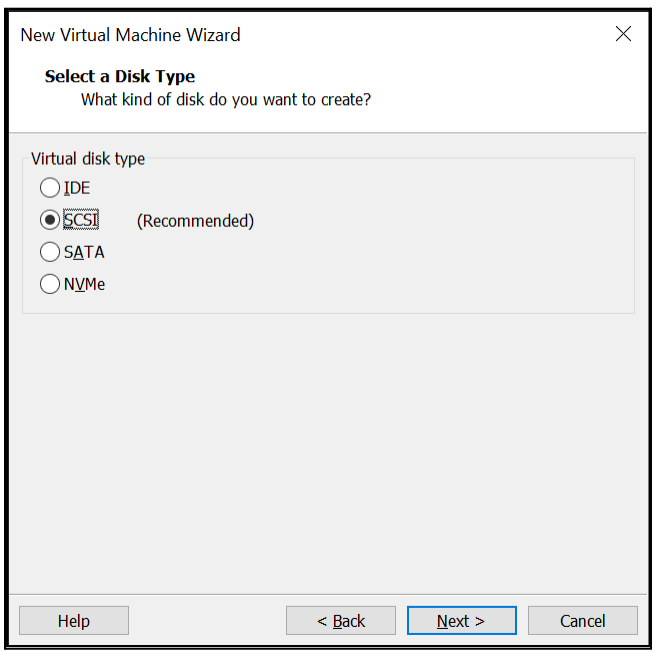


3. Configure Storage

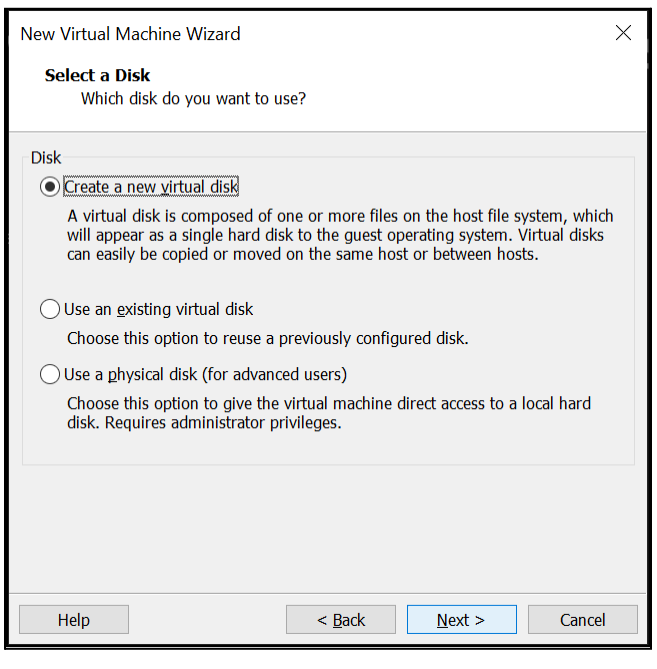
1. Keep the recommended I/O controller type and select **Next**.



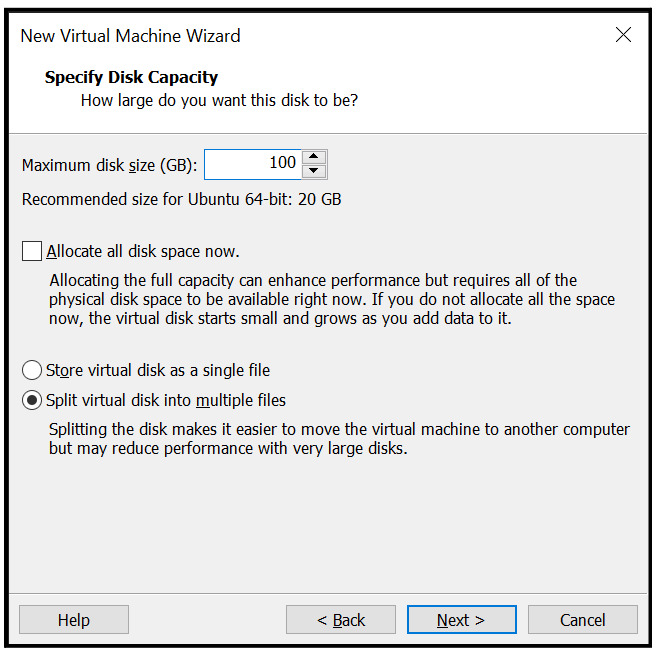
2. Keep the recommended disk type and select **Next**.



3. Select **Create a new virtual disk** and choose **Next**.

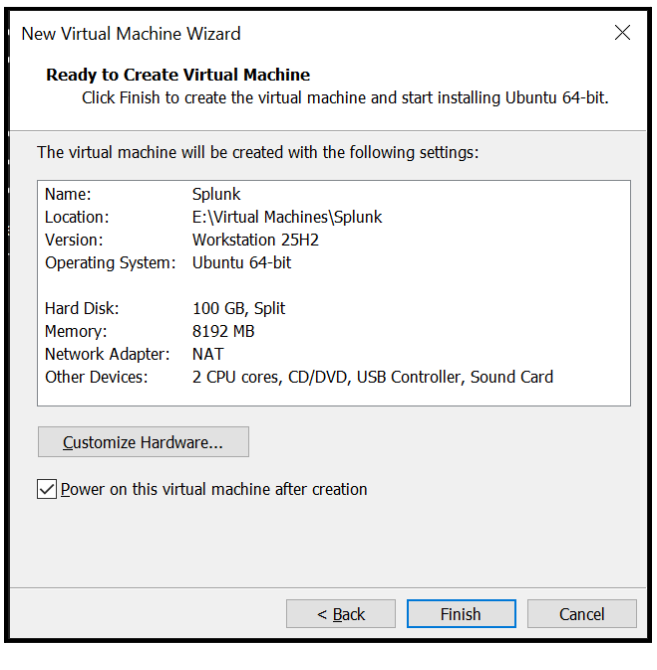


4. Set the maximum disk size to **100 GB**.



5. Leave the disk file name as default and select **Next**.

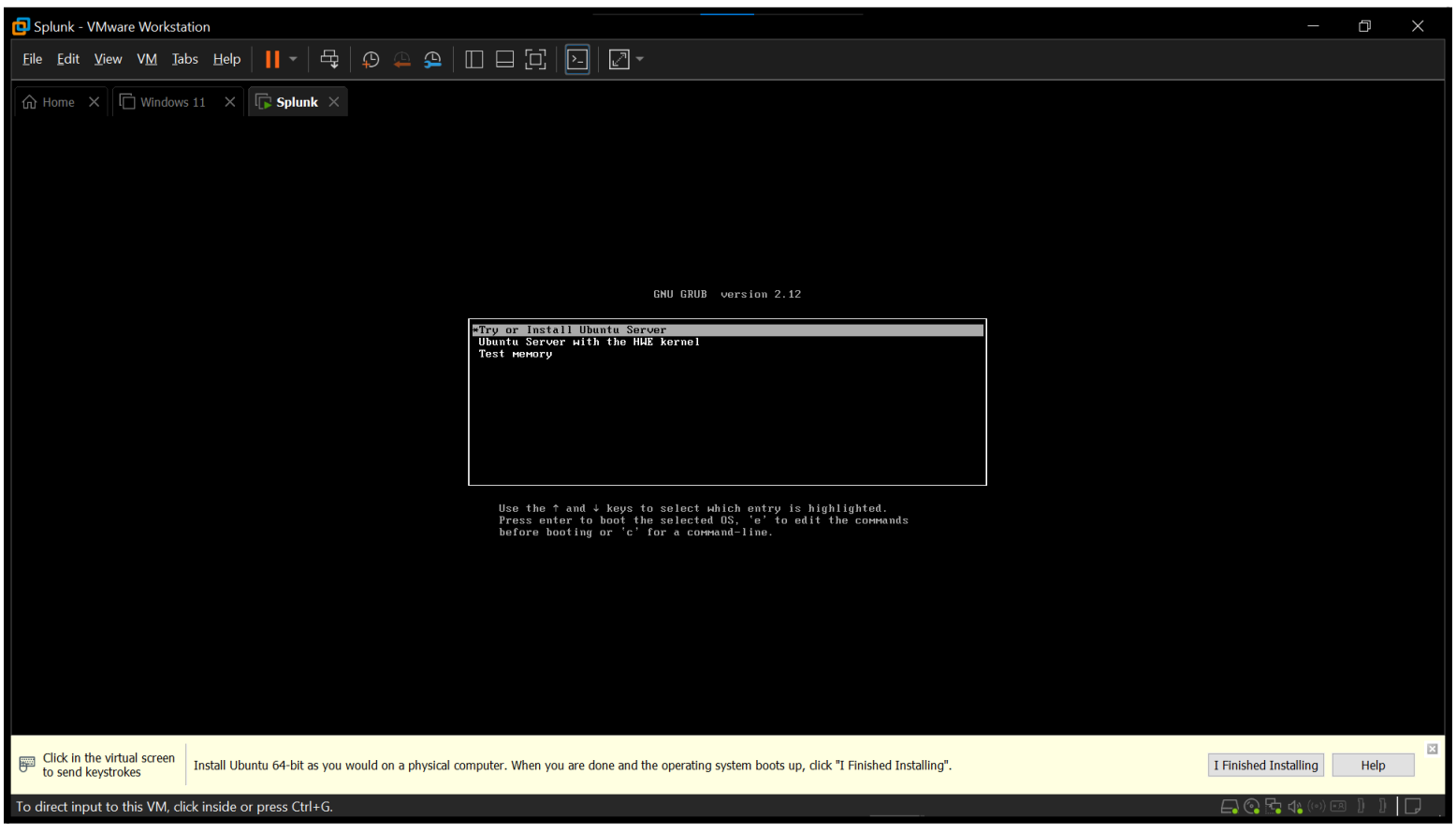
6. Check **Power on this virtual machine after creation** and select **Finish**.



The virtual machine will now boot from the ISO.

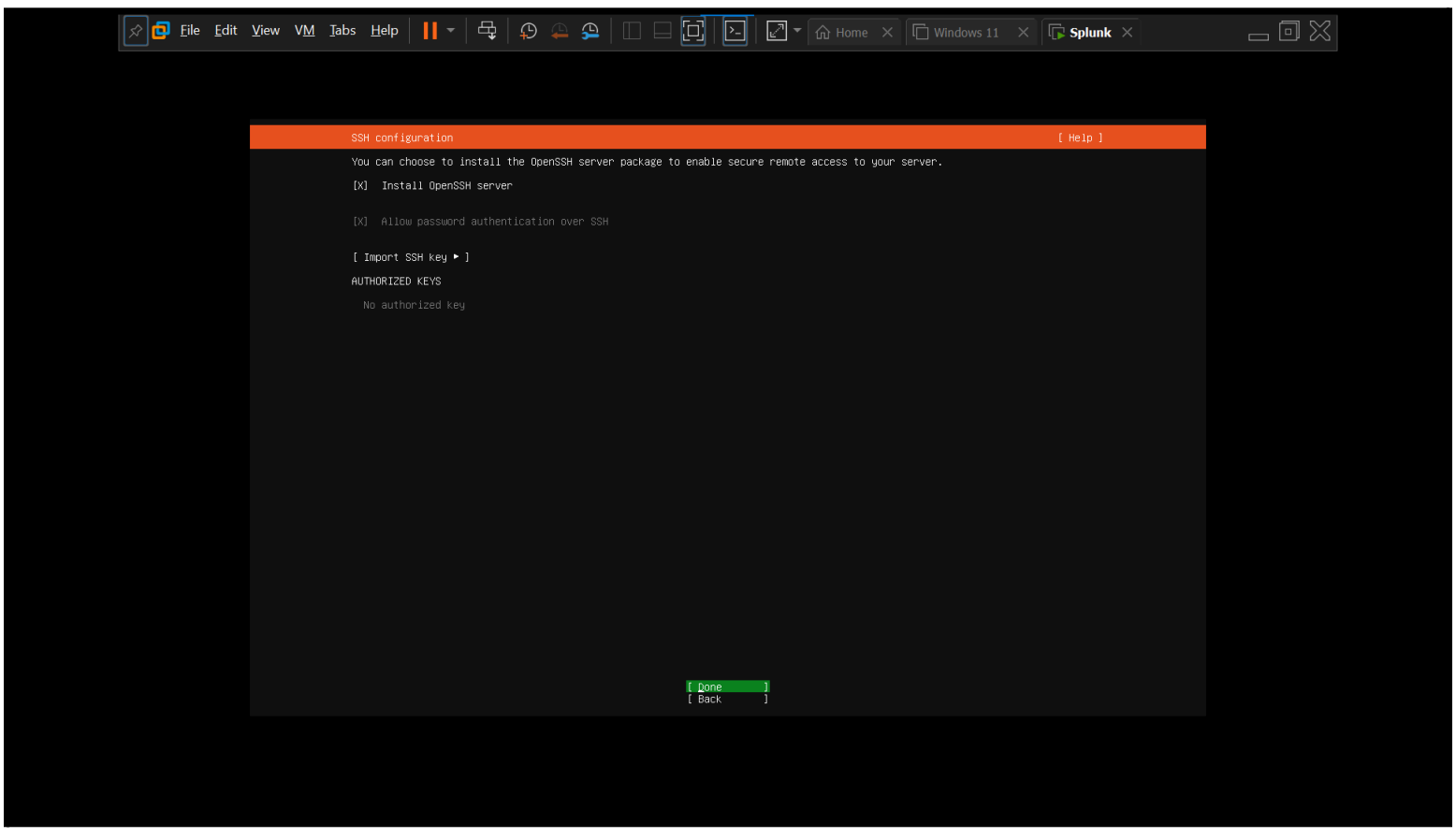
4. Install Ubuntu Server

1. On the boot menu, select **Try or Install Ubuntu Server** and press **Enter**.

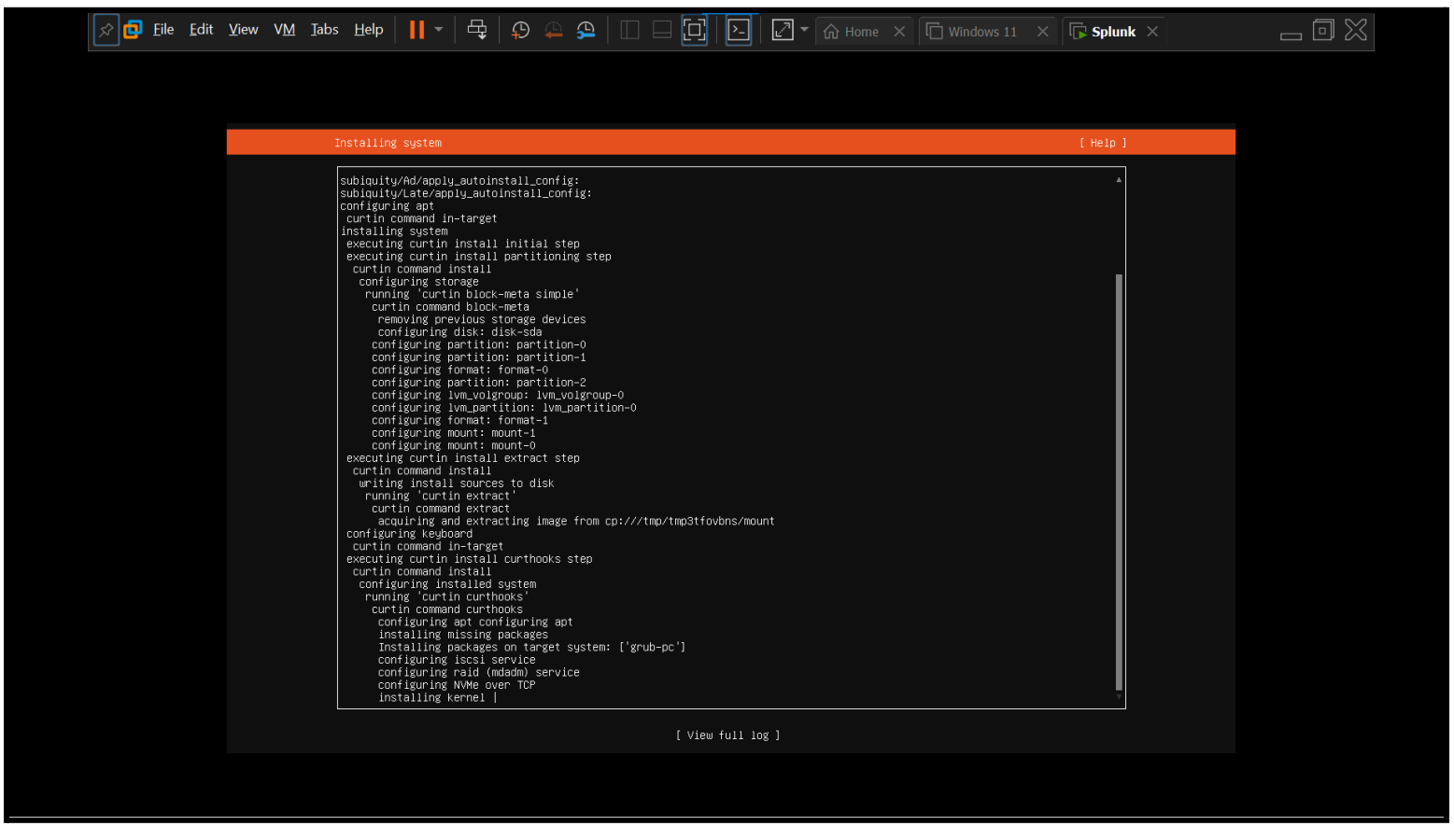


2. Choose your preferred language and continue.
3. Select **Done** on the default options until you reach the **Profile Configuration** screen.

4. Enter your full name, server name, username, and password and select **Done**.
5. When prompted for Ubuntu Pro, select **Skip for now**.
6. Enable **OpenSSH Server** when prompted.

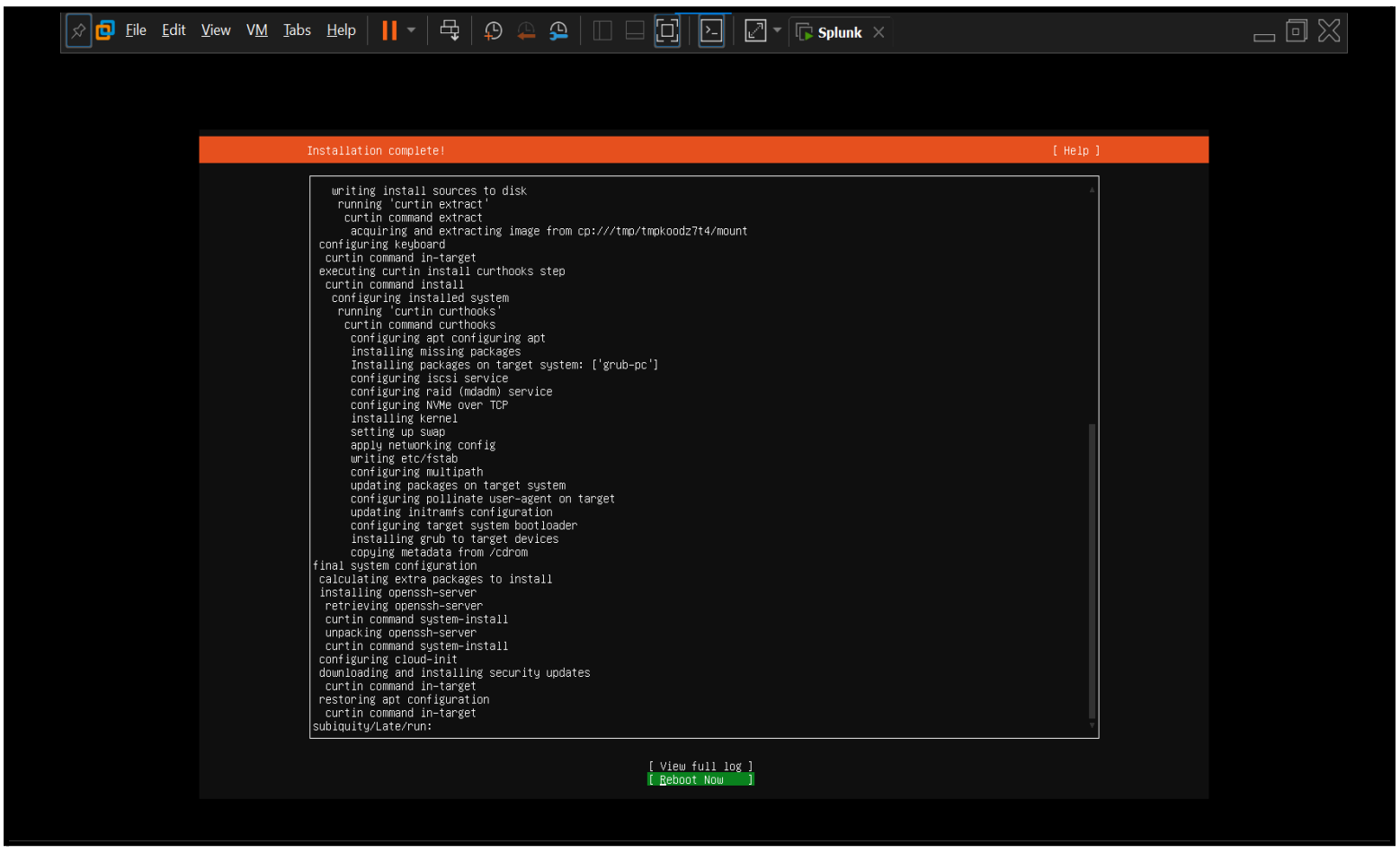


7. Select **Done** twice to start the installation.



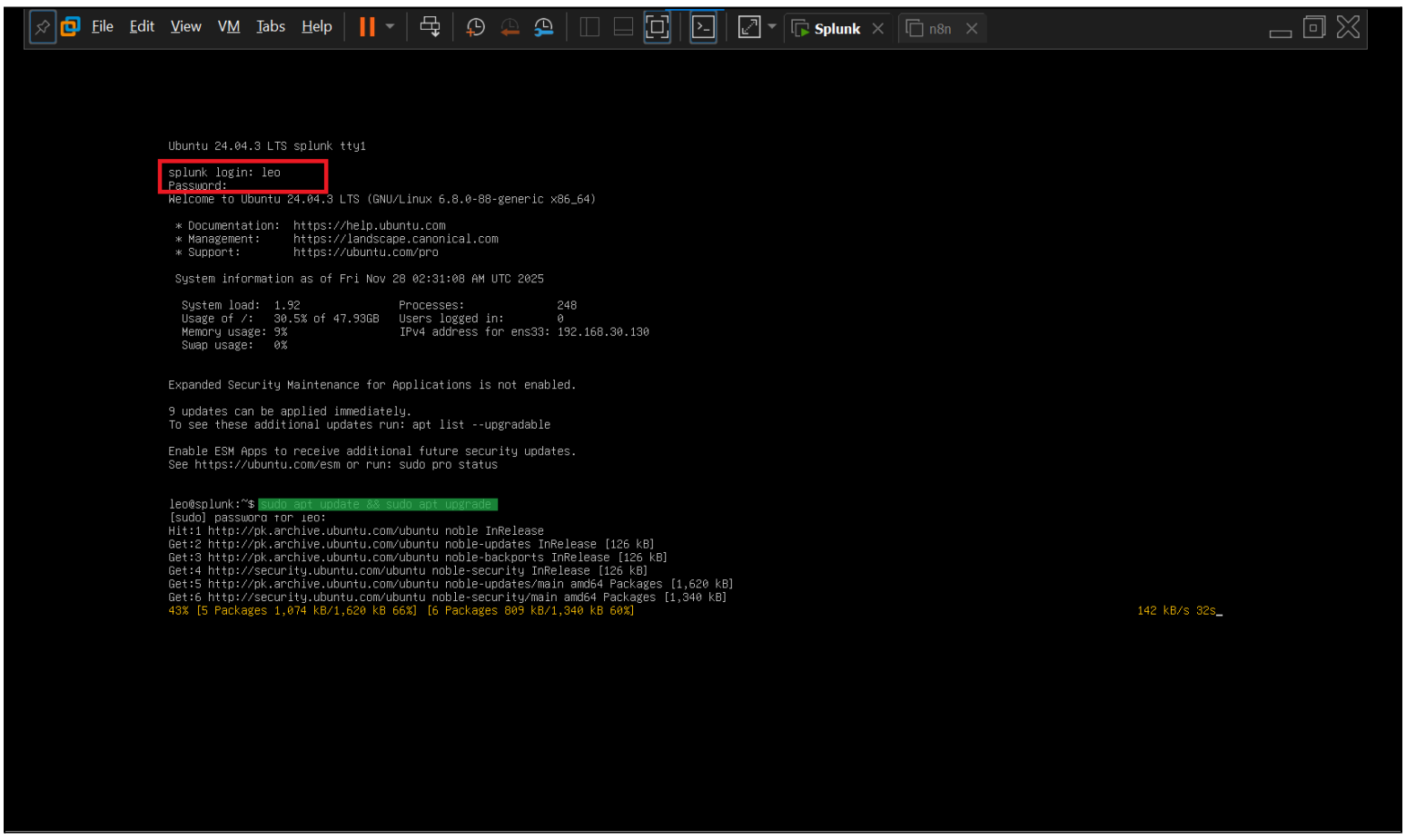
When finished:

1. Select **Reboot Now**.



2. Log in using the username and password you created earlier, then update the system packages by running:

`sudo apt update && sudo apt upgrade`

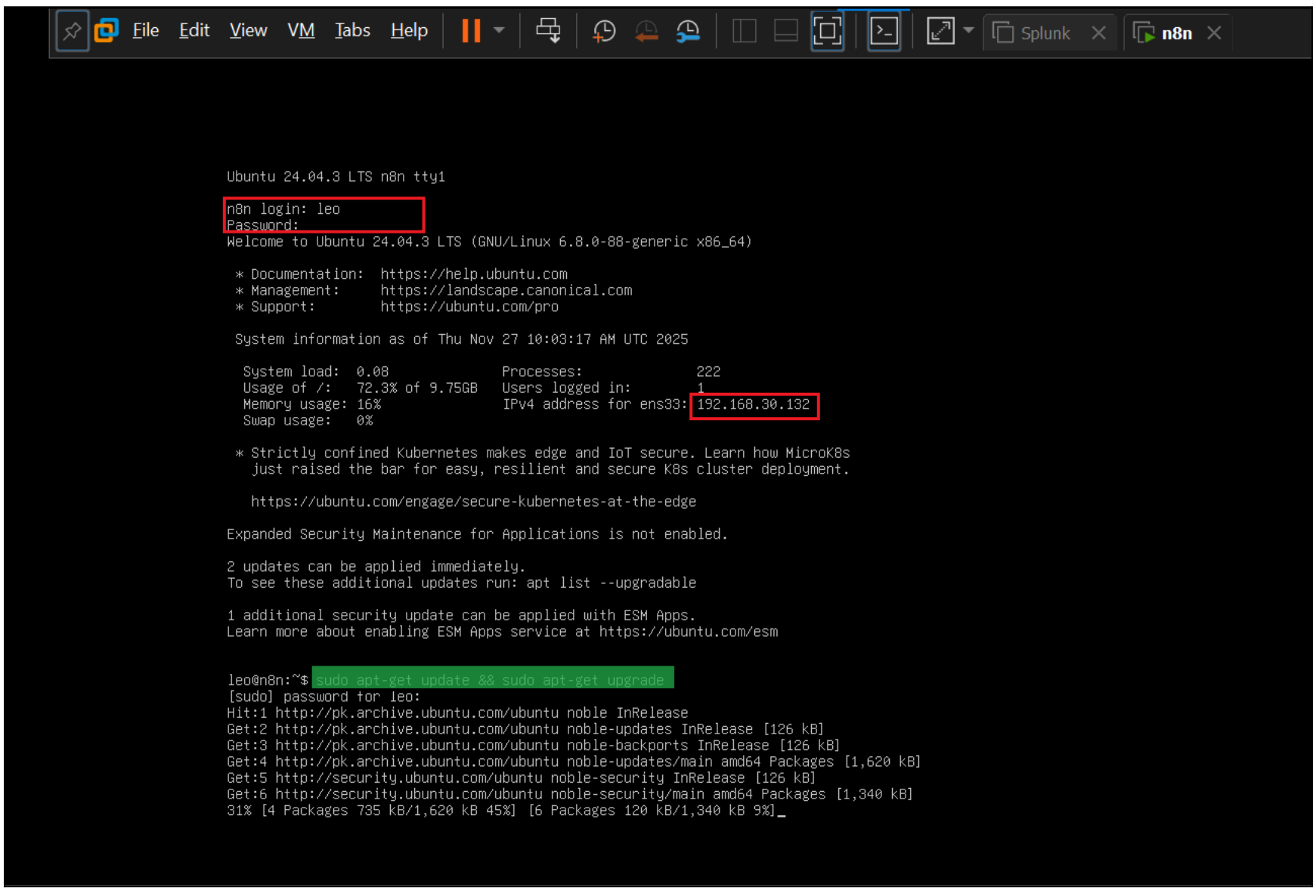


If the VM gets stuck at the “Installing system” stage, close the VM and start the installation again.

Next Step

You now have both the **Windows 11 VM** and the Ubuntu Server VM for **Splunk** set up. The only remaining VM you need is an Ubuntu Server instance for **n8n**. Follow the same VMware creation steps to create this VM.

After creating the **n8n VM**, update the system packages by running: `sudo apt update && sudo apt upgrade`



Once the **n8n VM** is ready, your lab environment will consist of three virtual machines:

- One **Windows 11 VM**
- Two Ubuntu Server VMs (one for **Splunk** and one for **n8n**)

Example IP addresses from my lab:

- Windows 11: **192.168.30.128**
- Splunk server: **192.168.30.130**
- n8n server: **192.168.30.132**

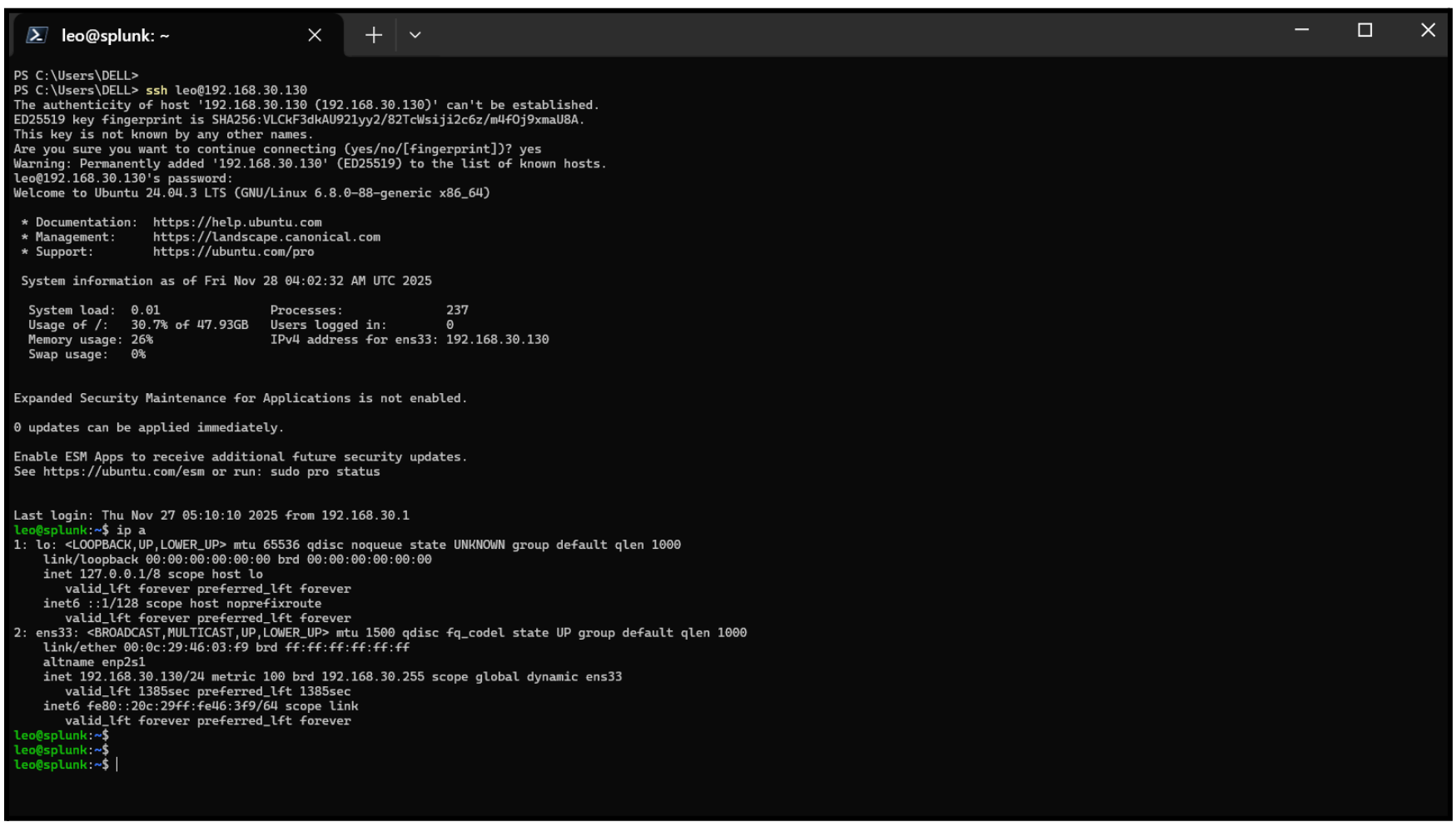
Your IP addresses will likely be different. Make sure to note them, as you will need them later.

Now that all VMs are set up, you can install Splunk on the Splunk VM.

Installing and Starting Splunk on the Splunk VM

Before installing, SSH into your Splunk VM to make it easier to copy and paste commands. From your host machine, run:

```
ssh {splunk-vm-username}@{your-splunk-vm-ip}
```



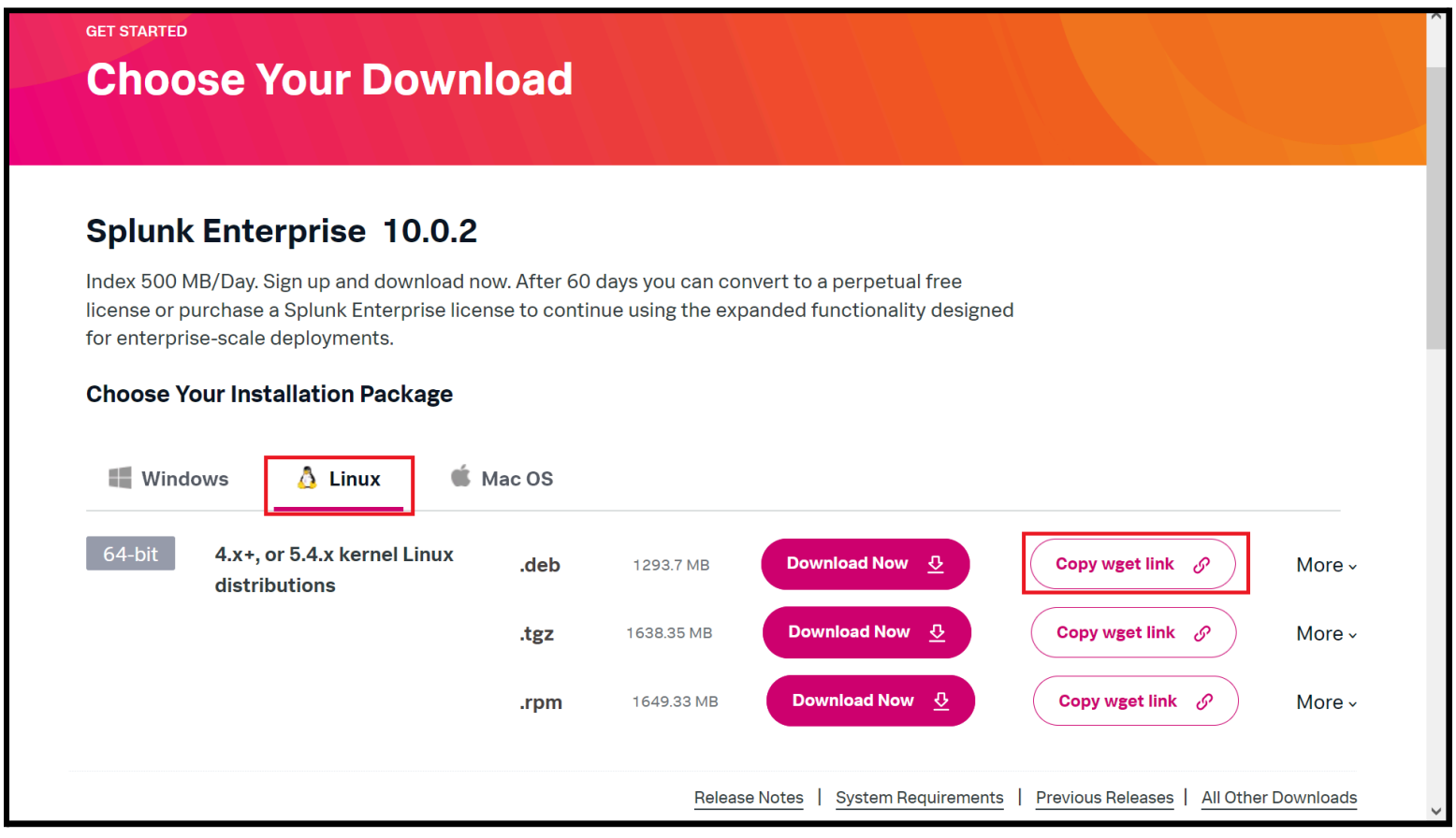
Type **yes** when prompted to confirm the connection.

- Enter the password for the Splunk VM user.

You will then be logged into the Splunk VM and ready to proceed with the installation.

Next, you need a Splunk account to download the software. If you don't have one, create an account on the Splunk website.

1. Go to **Free Trials & Downloads**.
2. Under **Splunk Enterprise**, click **Start Trial**.
3. Select **Linux**, then copy the **`wget`** link for the **`.deb`** package.



Return to your SSH session on the Splunk VM, paste the **`wget`** command, and press **Enter**. This will download the Splunk **`.deb`** package to your VM.

```
leo@splunk:~$ wget -O splunk-10.0.2-e2d18b4767e9-linux-amd64.deb "https://download.splunk.com/products/splunk/releases/10.0.2/linux/splunk-10.0.2-e2d18b4767e9-linux-amd64.deb"
```

1. **List the downloaded file** to confirm the download: `ls`
2. **Install the downloaded `.deb` package**: `sudo dpkg -i {downloaded-filename}`
    a. Wait until you see the **installation complete** message.
3. **Navigate to the Splunk directory**: `cd /opt/splunk/ && ll`

```
leo@splunk:~$ ls
splunk-10.0.2-e2d18b4767e9-linux-amd64.deb
leo@splunk:~$ sudo dpkg -i splunk-10.0.2-e2d18b4767e9-linux-amd64.deb
[sudo] password for leo:
Selecting previously unselected package splunk.
(Reading database ... 87325 files and directories currently installed.)
Preparing to unpack splunk-10.0.2-e2d18b4767e9-linux-amd64.deb ...
verify that this sytem has all the commands we will require to perform the preflight step
no need to run the splunk-preinstall upgrade check
Unpacking splunk (10.0.2) ...
Setting up splunk (10.0.2) ...
find: ‘/opt/splunk/lib/python3.7/site-packages’: No such file or directory
complete
leo@splunk:~$ cd /opt/splunk/ && ll
total 4520
drwxr-xr-x 11 splunk splunk    4096 Nov 23 13:34 ./
drwxr-xr-x  3 root   root      4096 Nov 23 13:32 ../
drwxr-xr-x  4 splunk splunk   12288 Nov 23 13:34 bin/
-r--r--r--  1 splunk splunk      57 Oct 29 23:44 copyright.txt
drwxr-xr-x 17 splunk splunk    4096 Nov 23 13:34 etc/
-rw-r--r--  1 splunk splunk     426 Nov 23 13:34 ftr
drwxr-xr-x  3 splunk splunk    4096 Nov 23 13:34 include/
drwxr-xr-x  8 splunk splunk    4096 Nov 23 13:34 lib/
-r--r--r--  1 splunk splunk   59708 Oct 29 23:44 license-eula.txt
-r--r--r--  1 splunk splunk    1090 Oct 29 16:29 LICENSE.txt
drwxr-xr-x  2 splunk splunk    4096 Nov 23 13:34 openssl/
drwxr-xr-x  7 splunk splunk    4096 Nov 23 13:34 opt/
drwxr-xr-x  2 splunk splunk    4096 Nov 23 13:34 quarantined_files/
-r--r--r--  1 splunk splunk     522 Oct 29 23:48 README-splunk.txt
drwxr-xr-x  6 splunk splunk    4096 Nov 23 13:34 share/
-r--r--r--  1 splunk splunk 4494014 Oct 30 00:17 splunk-10.0.2-e2d18b4767e9-linux-amd64-manifest
drwxr-xr-x  2 splunk splunk    4096 Nov 23 13:34 swidtag/
leo@splunk:/opt/splunk$
```

4. **Switch to the Splunk user and start Splunk**:

```
sudo -u splunk bash

 cd /opt/splunk/bin

 ./splunk start
```

a. Scroll through the license agreement and press **y**.

```
leo@splunk:/opt/splunk$ sudo -u splunk bash
splunk@splunk:~$ cd /opt/splunk/bin/
splunk@splunk:~/bin$ ./splunk start
Splunk General Terms (v4 August 2024)

These Splunk General Terms ("General Terms") between Splunk Inc., a Delaware
corporation, with its principal place of business at 250 Brannan Street, San
Francisco, California 94107, USA ("Splunk" or "we" or "us" or "our") and you
("Customer" or "you" or "your") govern your acquisition, access to, and use of
Splunk's Offerings, regardless of how accessed or acquired, whether directly
from us or from another Approved Source. By clicking on the appropriate button,
or by downloading, installing, accessing, or using any Offering, you agree to
these General Terms. If you are entering into these General Terms on behalf of
Customer, you represent that you have the authority to bind Customer. If you do
not agree to these General Terms, or if you are not authorized to accept the
General Terms on behalf of Customer, do not download, install, access, or use
any Offering. The "Effective Date" of these General Terms is: (i) the date of
Delivery; or (ii) the date you access or use the Offering in any way, whichever
is earlier. Capitalized terms are defined in the Definitions section below.
Effective September 23, 2024, and unless the context otherwise requires, any
reference in these General Terms to "Splunk Inc.", "Splunk", "we", "us" or "our"
will be deemed to refer to "Splunk LLC".

1. Your Use Rights and Limits

1.1. Your Use Rights. We grant you a non-exclusive, worldwide, non-transferable
and non-sublicensable right, subject to your compliance with these General Terms
and payment of applicable Fees, to use acquired Offerings only for your Internal
Business Purpose during the Term, up to the Capacity, and, if applicable, in
accordance with the Order ("Use Rights"). You have the right to make a
reasonable number of copies of On-Premises Products for archival and back-up
purposes.

1.2. Limits on Your Use Rights. Except as expressly permitted in the Order,
these General Terms or Documentation, your Use Rights exclude the right to, and
you agree not to (nor allow any user or Third Party Provider to): (i) reverse
```

b. Set a **username and password** for Splunk login. These credentials will be used to access the Splunk dashboard.

```
Please enter an administrator username: leo
Password must contain at least:
   * 8 total printable ASCII character(s).
Please enter a new password:
Please confirm new password:
Copying '/opt/splunk/etc/openldap/ldap.conf.default' to '/opt/splunk/etc/openldap/ldap.conf'.
writing RSA key

writing RSA key

Moving '/opt/splunk/share/splunk/search_mrsparkle/modules.new' to '/opt/splunk/share/splunk/search_mrsparkle/modules'.

Splunk> Finding your faults, just like mom.

Checking prerequisites...
        Checking http port [8000]: open
        Checking mgmt port [8089]: open
        Checking appserver port [127.0.0.1:8065]: open
        Checking kvstore port [8191]: open
        Checking configuration... Done.
                Creating: /opt/splunk/var/lib/splunk
                Creating: /opt/splunk/var/run/splunk
                Creating: /opt/splunk/var/run/splunk/appserver/i18n
                Creating: /opt/splunk/var/run/splunk/appserver/modules/static/css
                Creating: /opt/splunk/var/run/splunk/upload
                Creating: /opt/splunk/var/run/splunk/search_telemetry
                Creating: /opt/splunk/var/run/splunk/search_log
                Creating: /opt/splunk/var/spool/splunk
                Creating: /opt/splunk/var/spool/dirmoncache
                Creating: /opt/splunk/var/lib/splunk/authDb
                Creating: /opt/splunk/var/lib/splunk/hashDb
                Creating: /opt/splunk/var/run/splunk/collect
                Creating: /opt/splunk/var/run/splunk/sessions
New certs have been generated in '/opt/splunk/etc/auth'.
New certs have been generated in '/opt/splunk/etc/auth'.
        Checking critical directories...        Done
```

5. **Exit the Splunk user shell**: `exit`
6. **Enable Splunk to start on boot and start the Splunk service:**

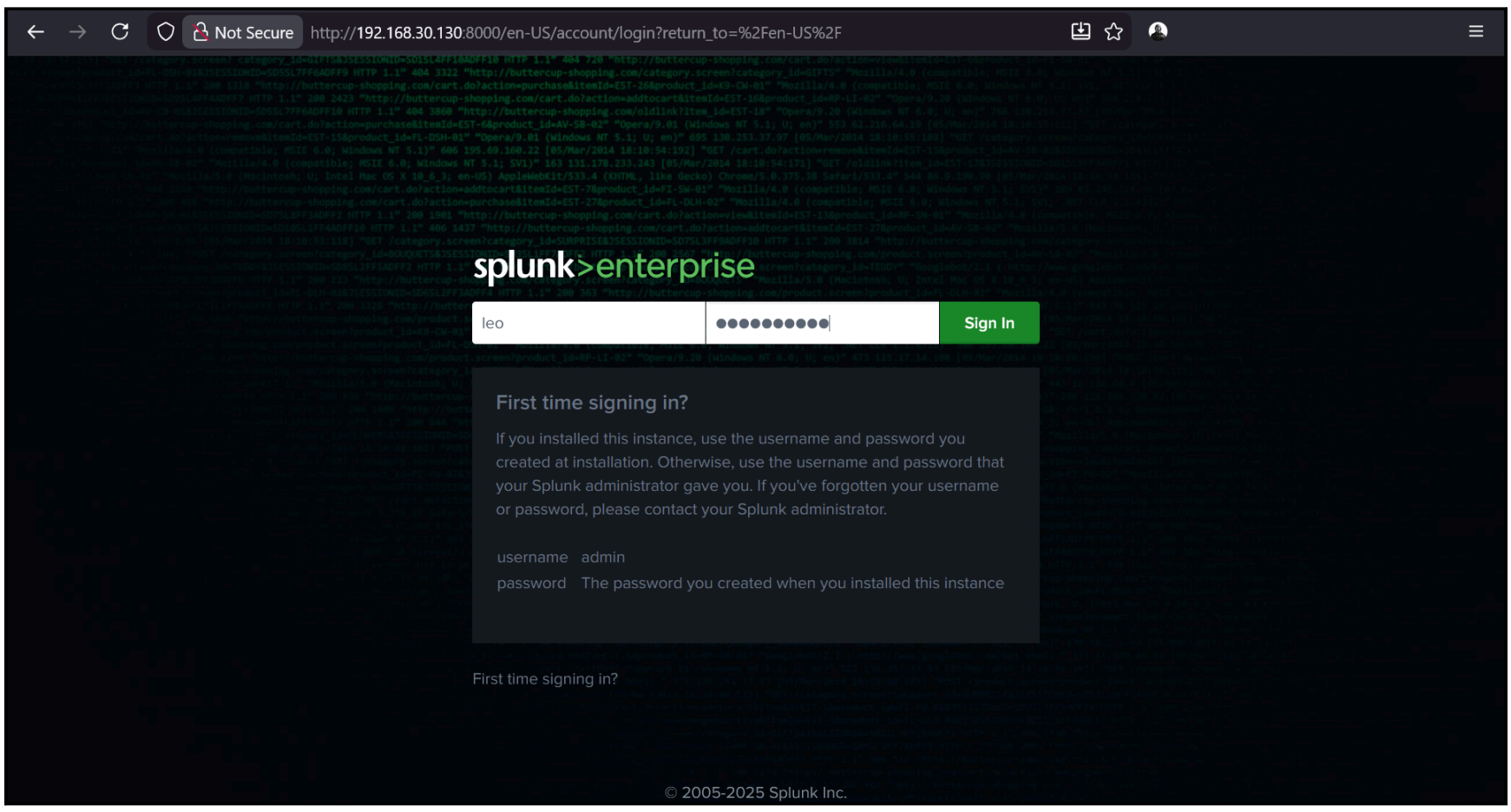
```
cd /opt/splunk/bin
sudo ./splunk enable boot-start -user splunk
sudo systemctl start splunk
sudo systemctl status splunk
```

```
The Splunk web interface is at http://splunk:8000

splunk@splunk:~/bin$ exit
exit
leo@splunk:/opt/splunk$ cd bin/
leo@splunk:/opt/splunk/bin$ sudo ./splunk enable boot-start -user splunk
Init script installed at /etc/init.d/splunk.
Init script is configured to run at boot.
leo@splunk:/opt/splunk/bin$
```

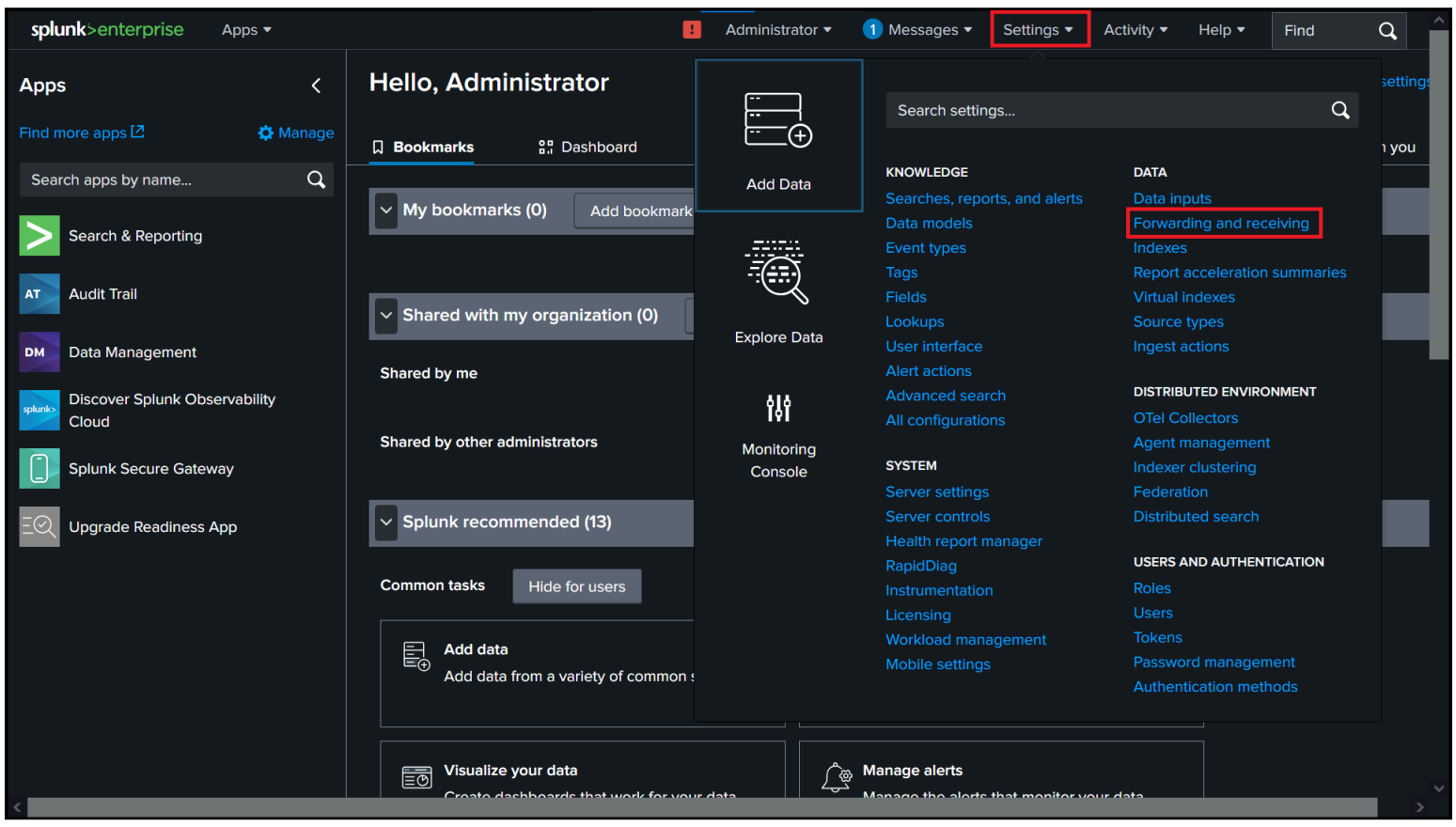
7. You should now have Splunk installed, running, and configured to start automatically on boot. The status command will confirm that the service is active.
8. Now, open a browser on your host machine and go to: http://{your-splunk-vm-ip}:8000

You should see the Splunk login page. Enter the username and password you created during setup and press **Enter** to access the Splunk dashboard.

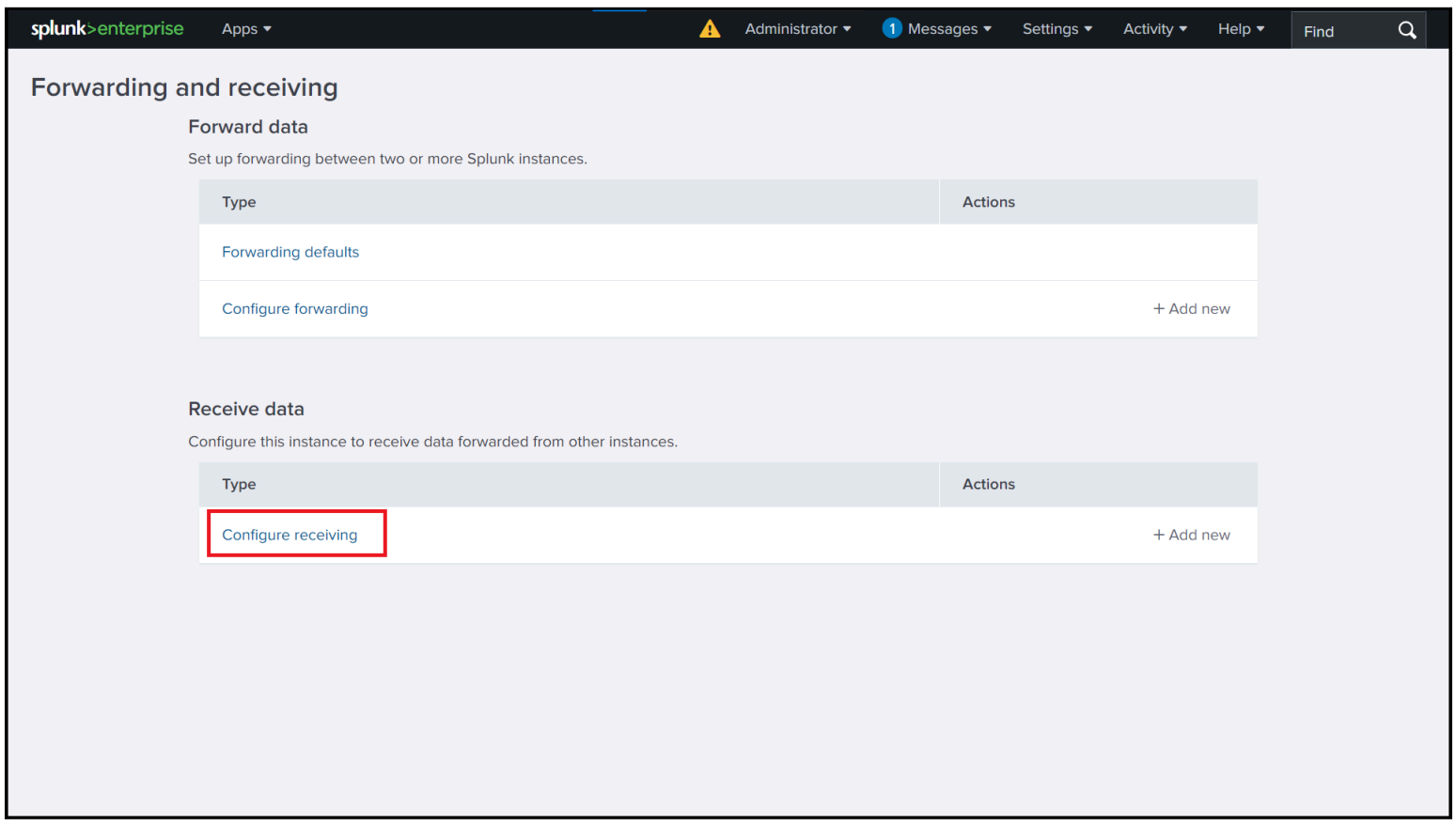


Configuring Splunk to Receive Logs

1. **Enable Receiving on Splunk**
   - In Splunk Web, go to **Settings > Forwarding and Receiving**.



- Under **Receiving Data**, click **Configure Receiving**.

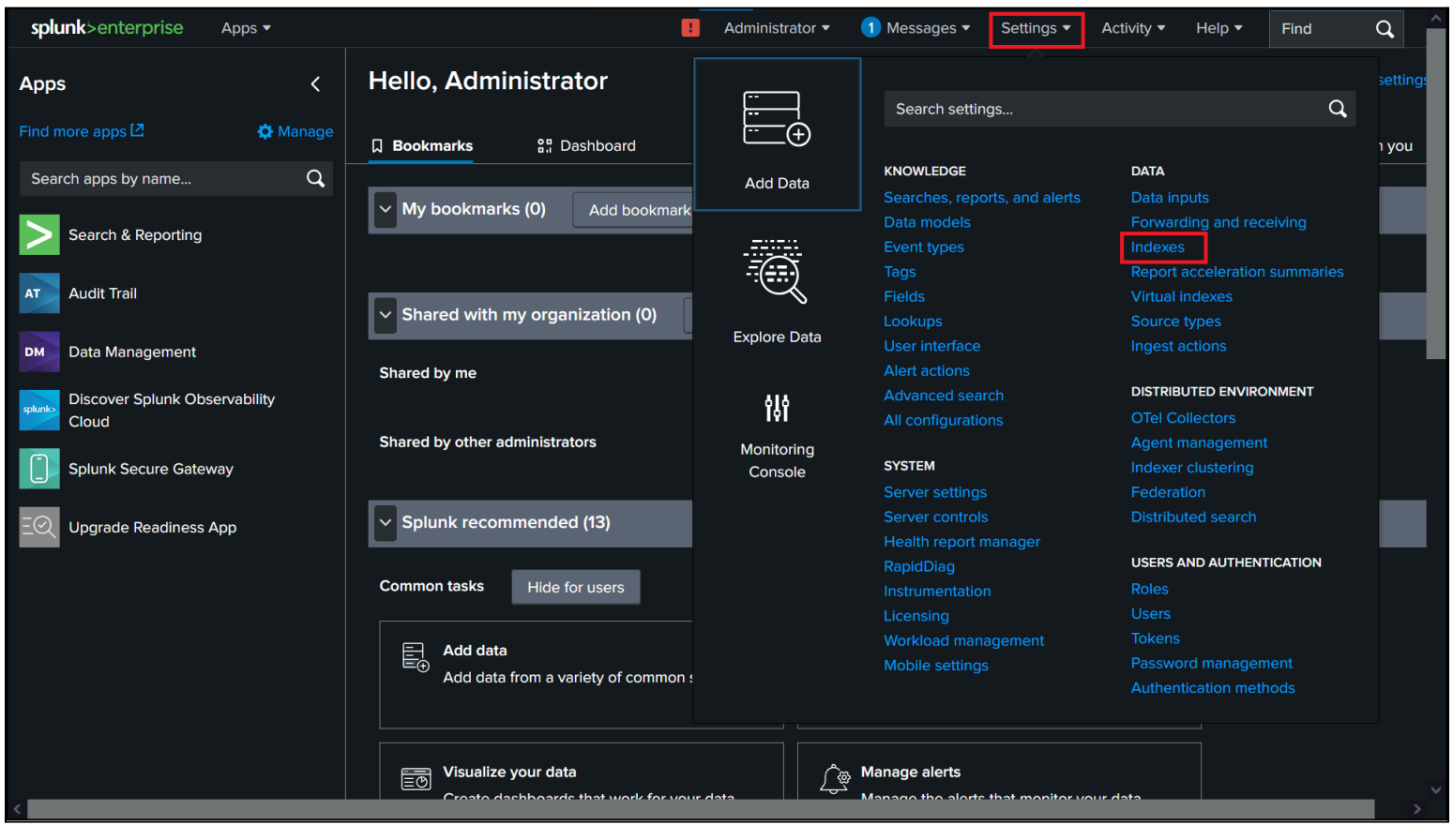


- Select **New Receiving Port**.

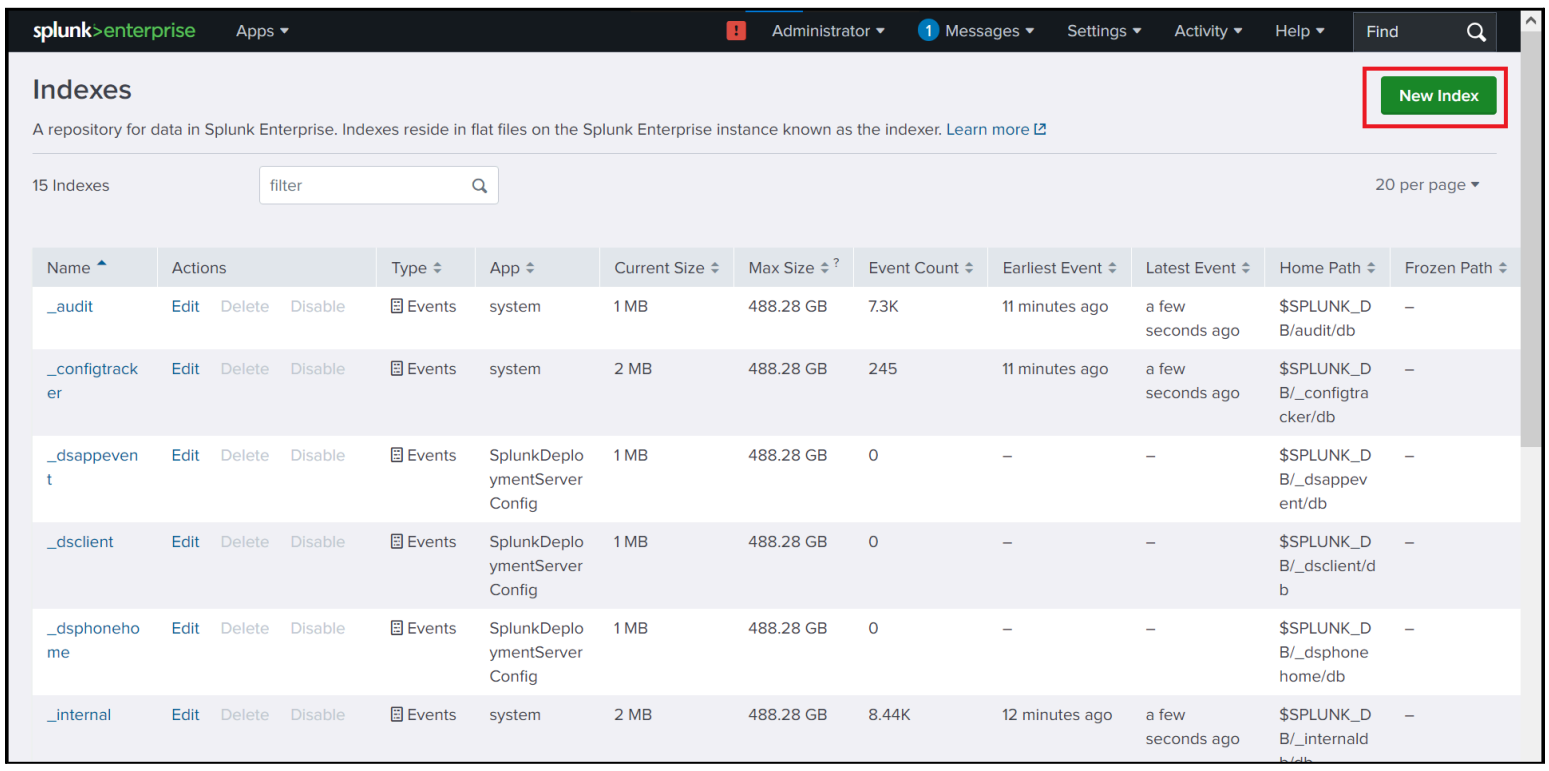
- Enter **9997** (the default port for Splunk forwarder traffic) and click **Save**.

2. **Create a New Index**

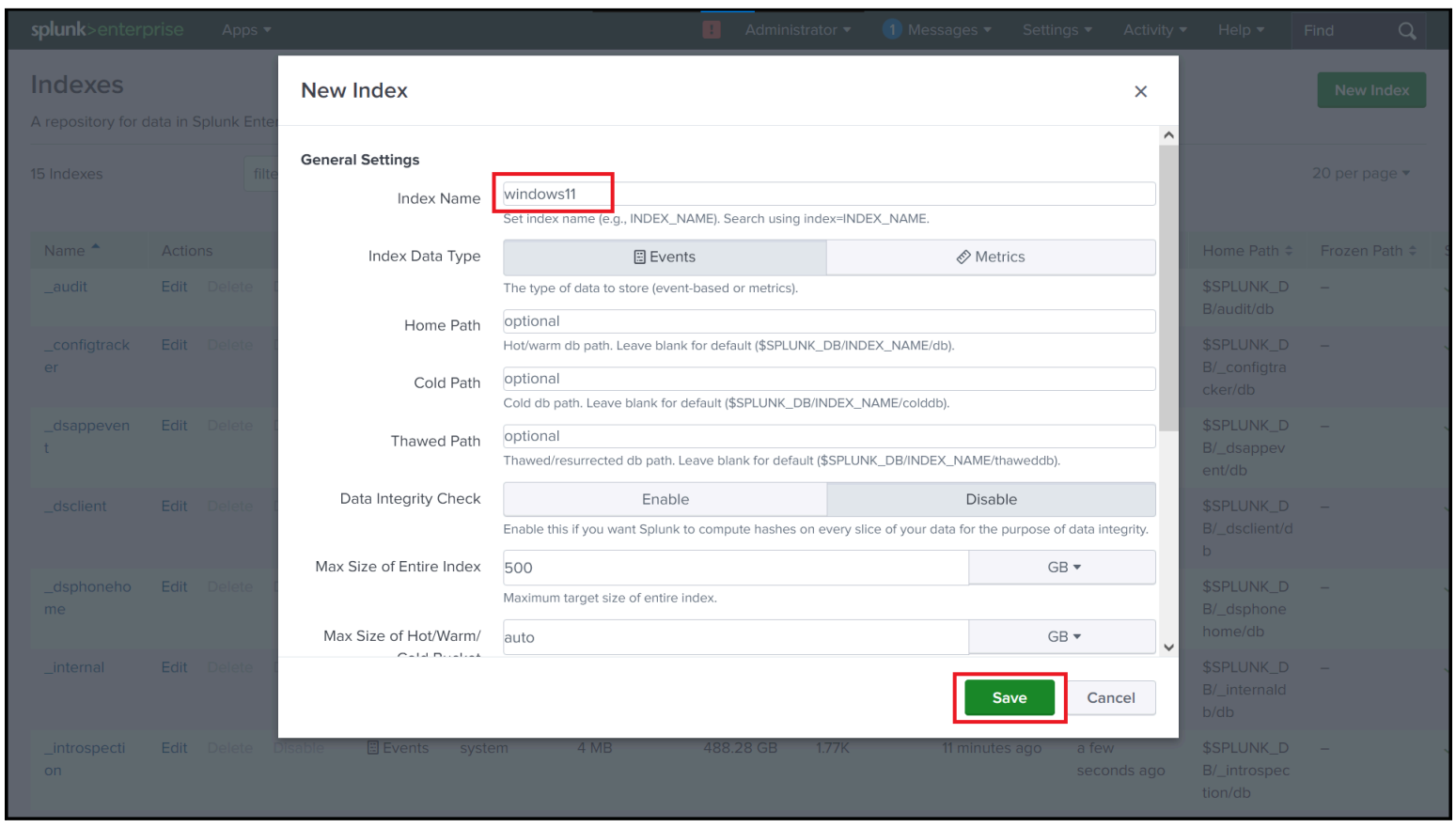
- Go to **Settings > Indexes**.



- Click **New Index**.

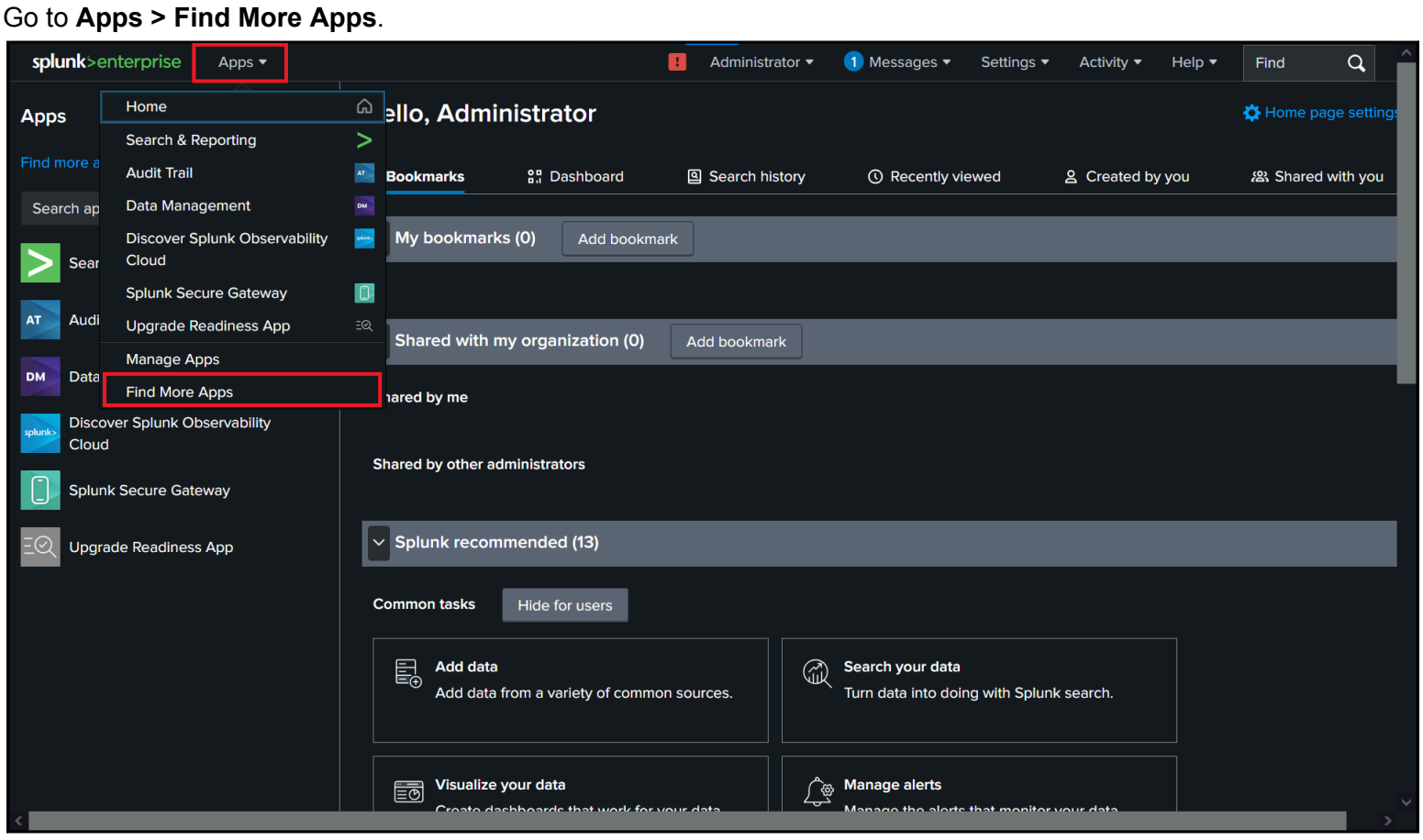


- Enter an index name (example: **windows11**) and click **Save**.



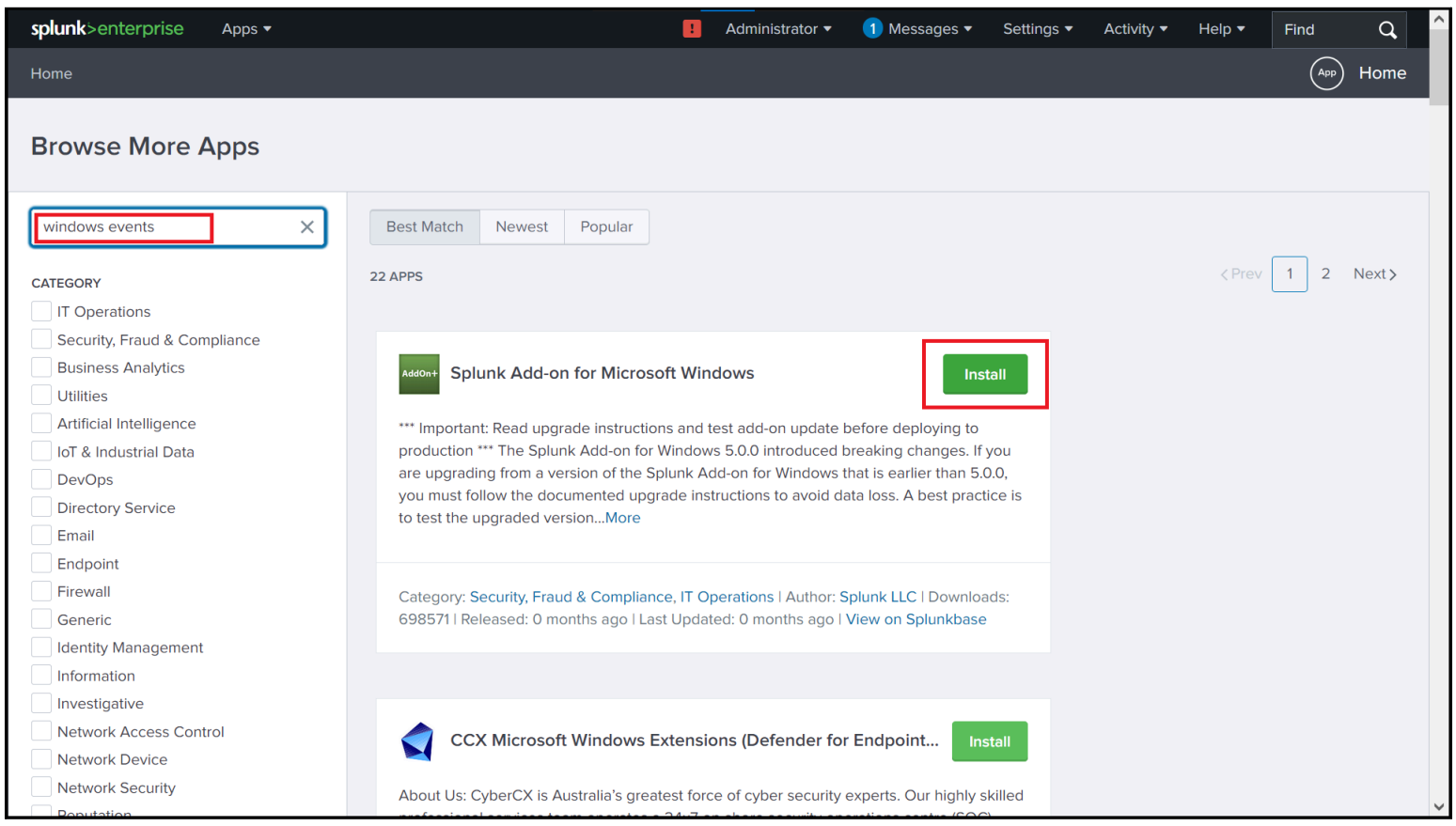
3. **Install the Splunk Add-on for Windows**

- Go to **Apps > Find More Apps**.



- Search for **Windows Events**.

- Select **Splunk Add-on for Microsoft Windows** and click **Install**.



- Sign in using your **Splunk.com account credentials** (these are the credentials for the Splunk website, not your Splunk dashboard login).

- Accept the terms and complete the installation.

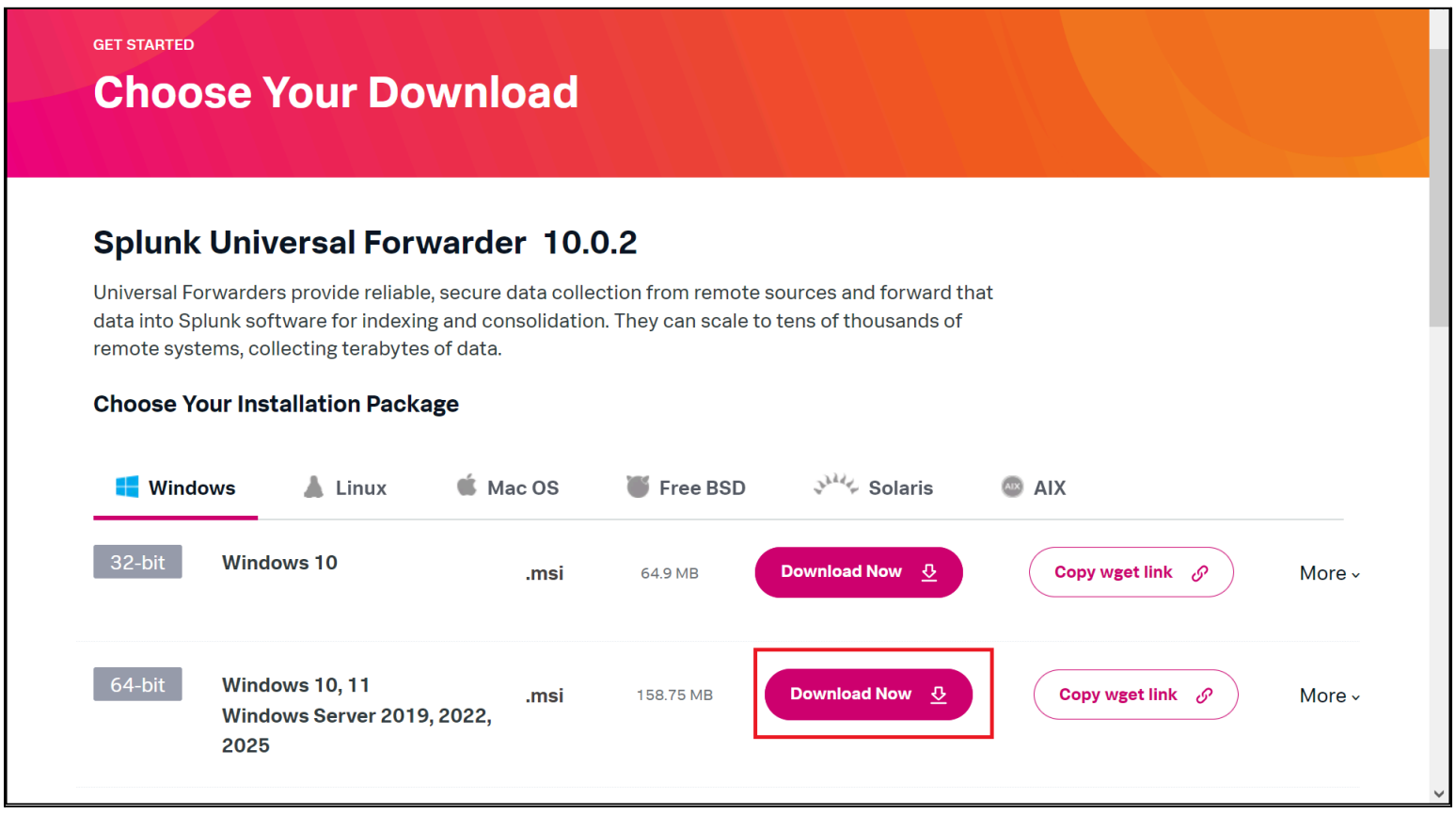
From the Windows 11 VM, open **Command Prompt**.
Ping the Splunk server: `ping <splunk-vm-ip>`

If you receive replies, network connectivity is working.

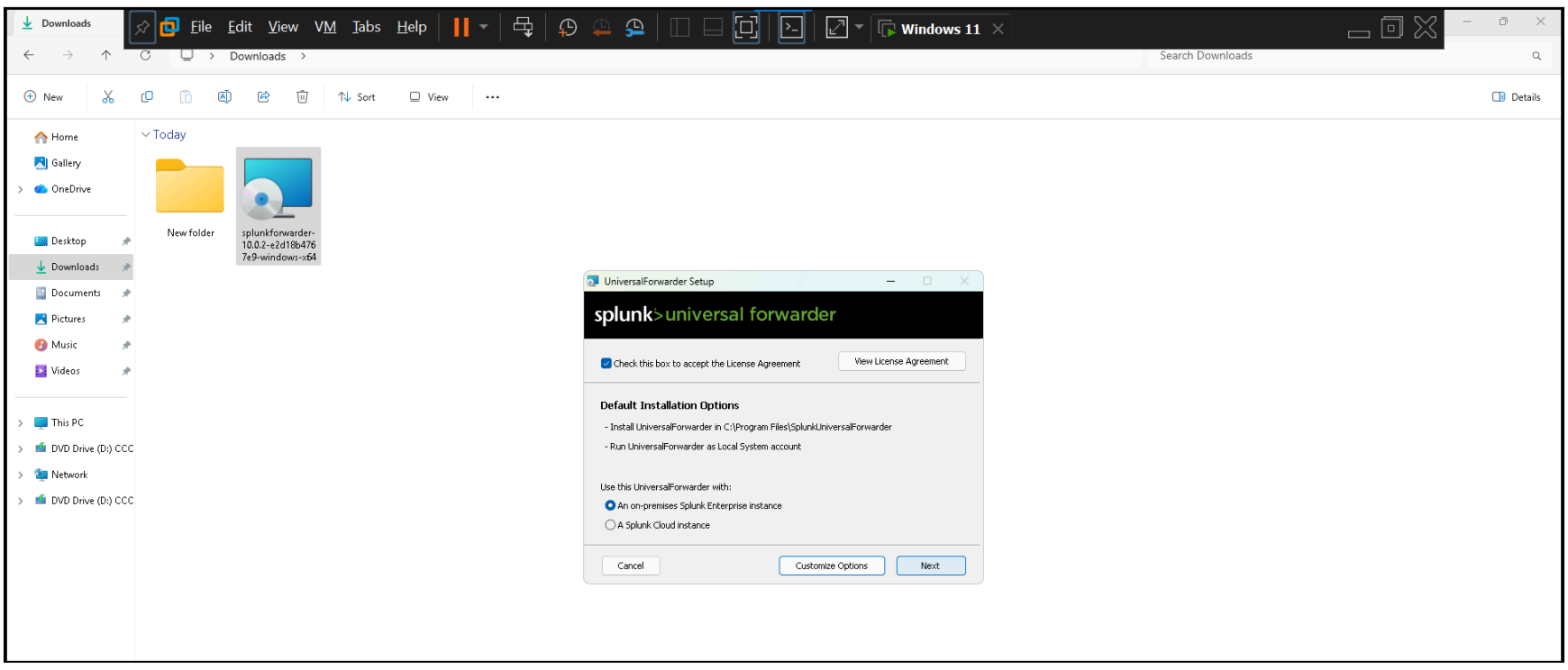
Download and Install the Splunk Universal Forwarder

- On the **Windows 11 VM**, open a web browser and go to the **Splunk website**.

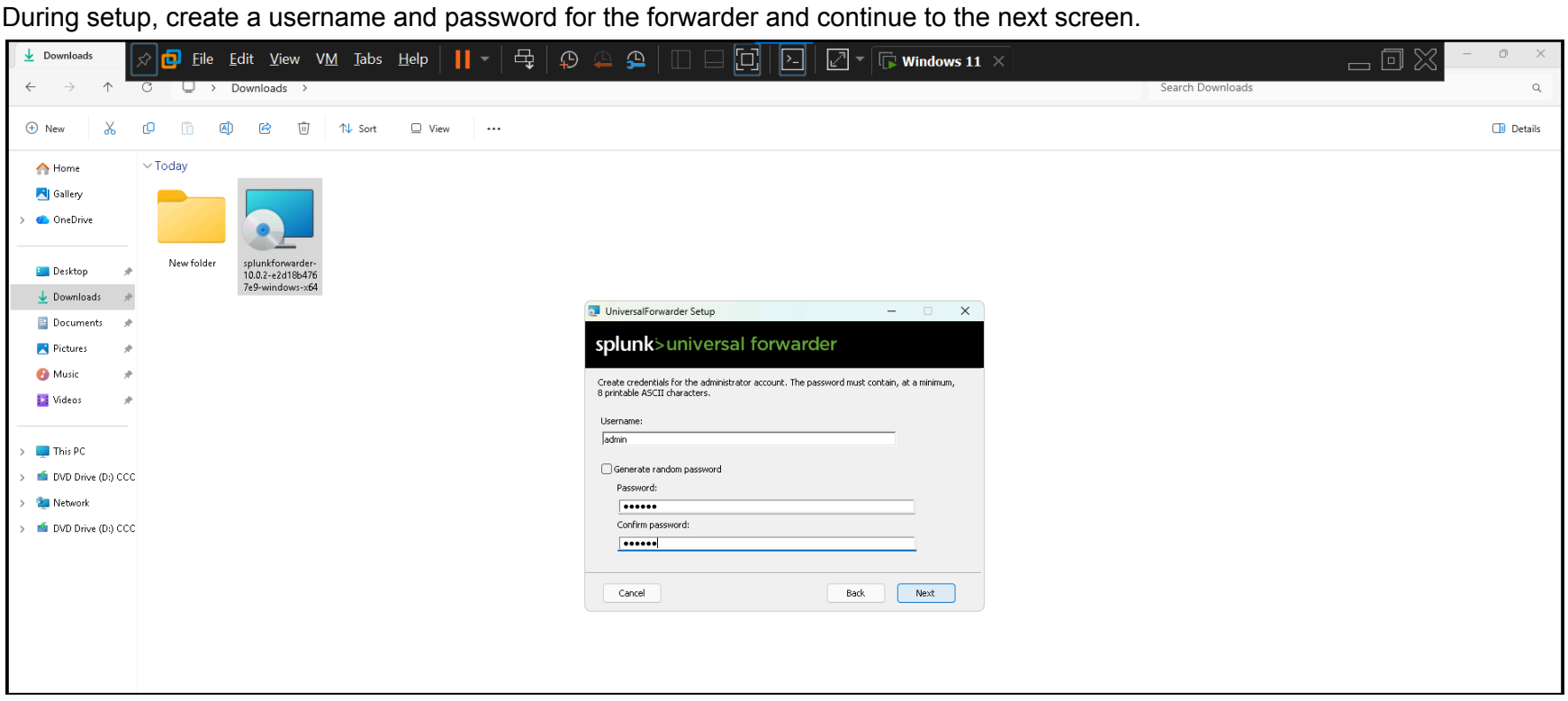
- Download the **Splunk Universal Forwarder for Windows11**.



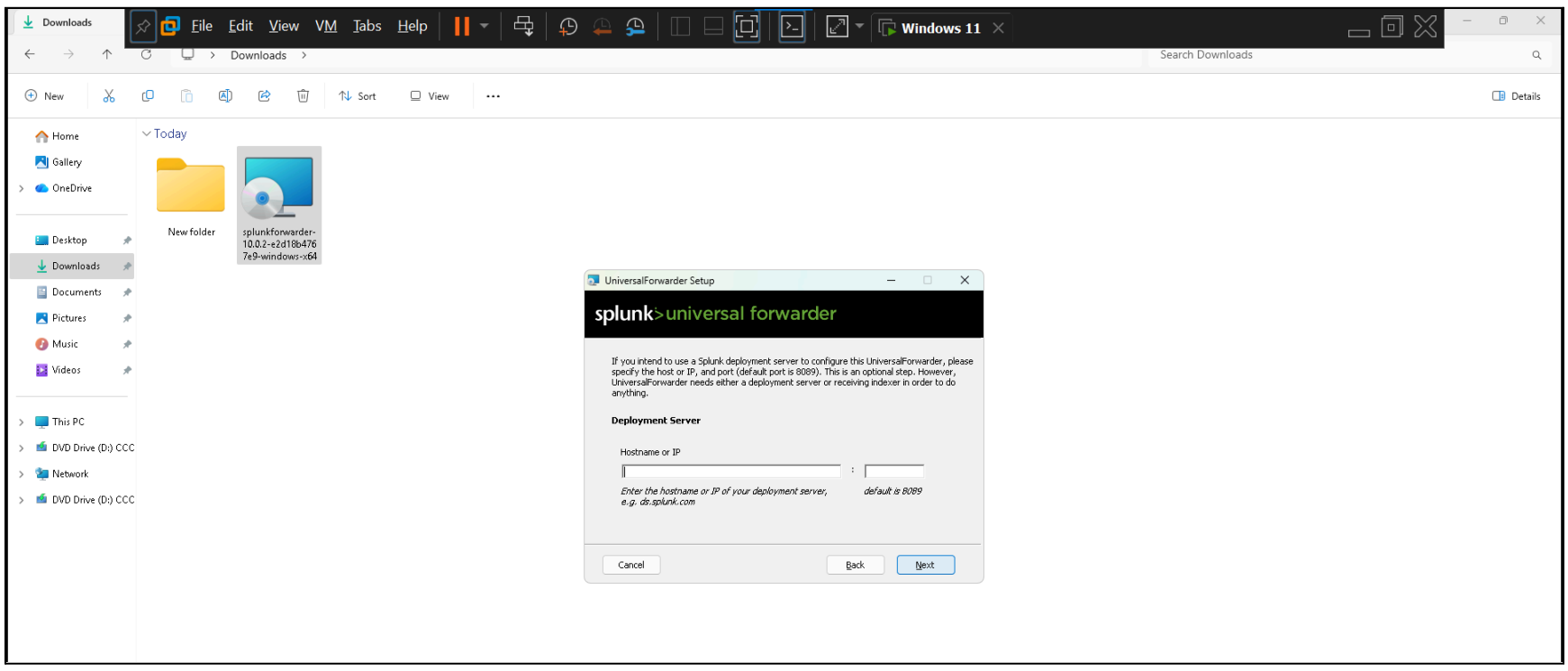
Now that the Universal Forwarder installer is inside the Windows 11 VM, double-click the file to begin the installation.



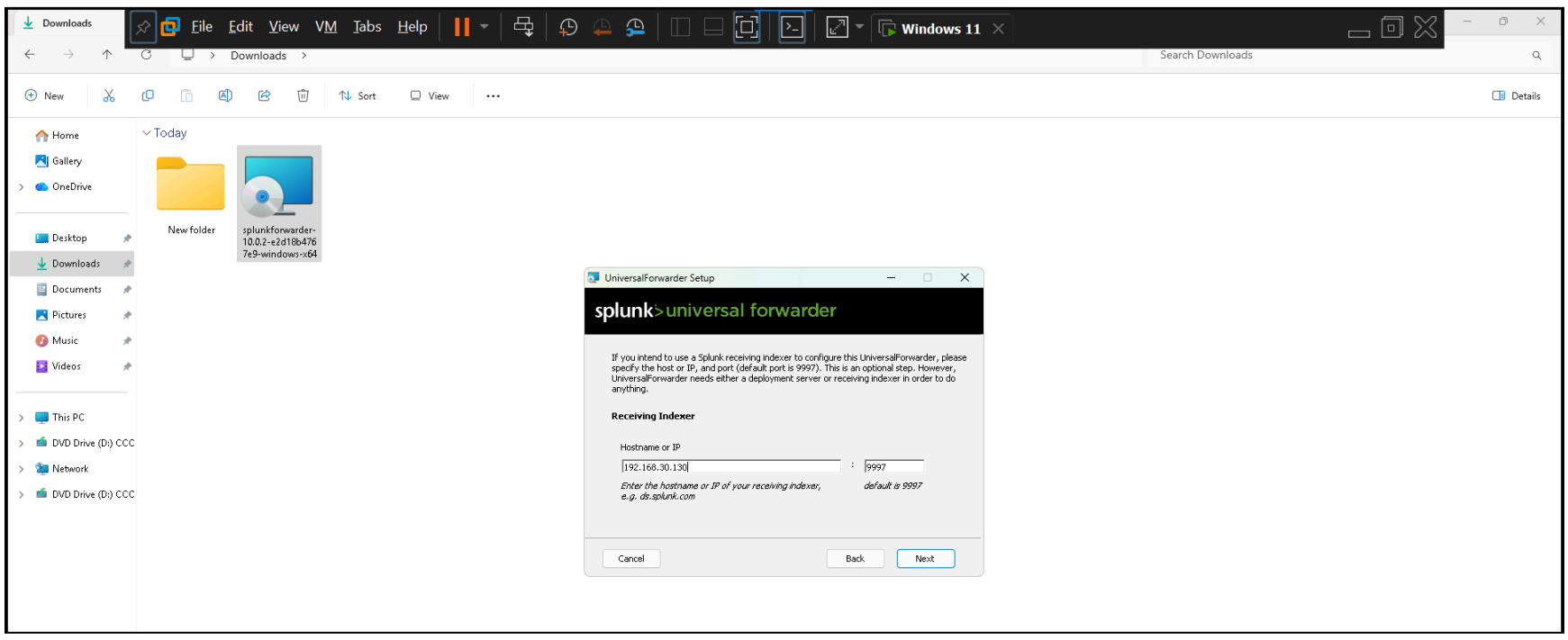
During setup, create a username and password for the forwarder and continue to the next screen.



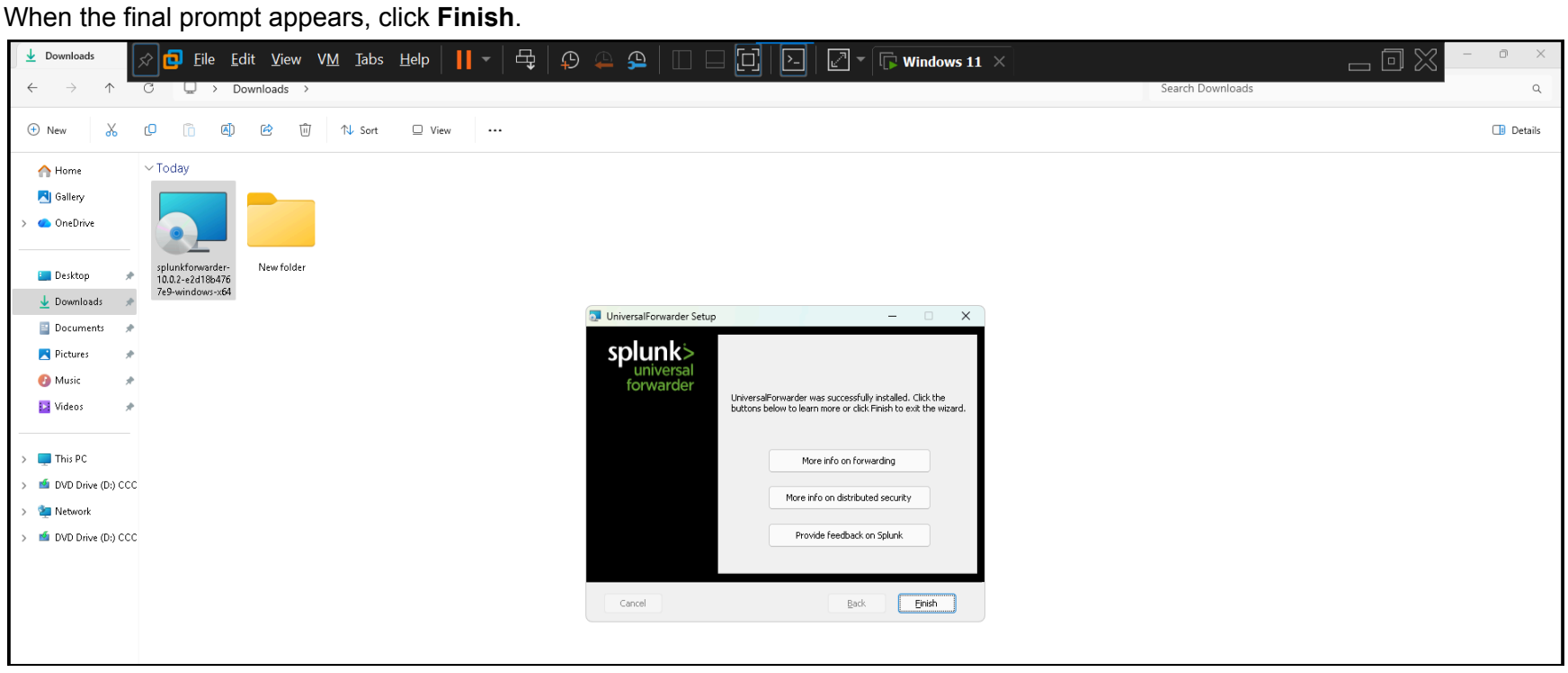
Skip the deployment server option and proceed.



Enter the **Splunk VM** IP address, specify the default receiving port **9997**, continue through the installer, and complete the installation.



When the final prompt appears, click **Finish**.



Creating and Configuring `inputs.conf`

The last configuration step on the Windows 11 VM involves the **inputs.conf** file, located at:

```
C:\Program Files\SplunkUniversalForwarder\etc\system\local
```

This file does not exist by **default**, so you need to create it manually.
On the Universal Forwarder, **inputs.conf** is the primary configuration file that defines what telemetry the forwarder should collect and send to Splunk. In simple terms, this file tells the forwarder:

> **“Collect this data, apply these settings (sourcetype, index, etc.), and forward it to the indexer.”**

You can define all the Windows telemetry you want to send inside this file. Any changes you make to **inputs.conf** typically require a restart of the Splunk Universal Forwarder service to take effect.

inputs.conf (Recommended Configuration)

Below is the full inputs.conf configuration used for forwarding Windows event logs to Splunk.
All logs are sent to the **windows11** index. If your index name differs, update the value accordingly.

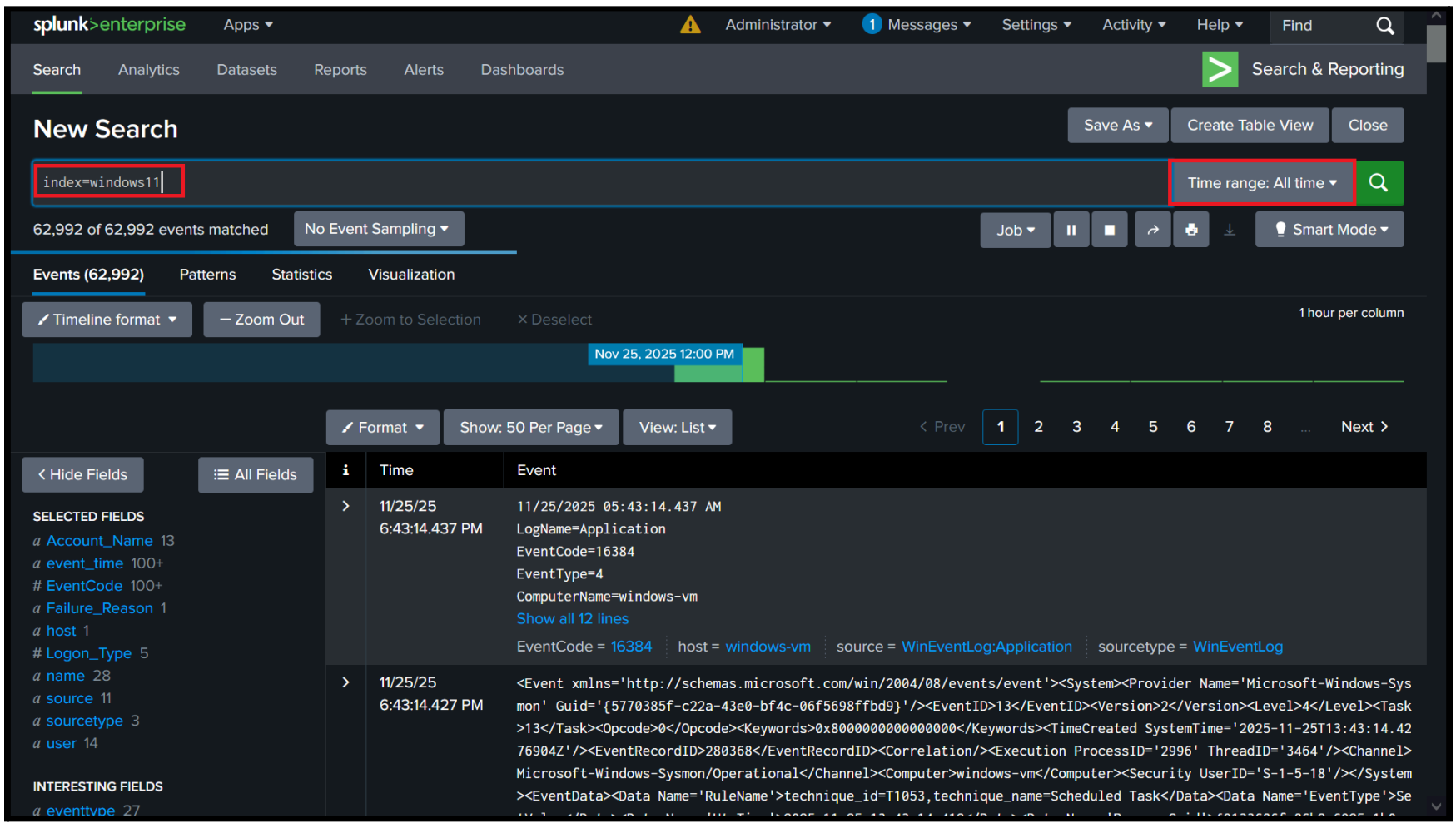
```
########## Sysmon ##########
[WinEventLog://Microsoft-Windows-Sysmon/Operational]
index = windows11
disabled = false
renderXml = true
source = XmlWinEventLog:Microsoft-Windows-Sysmon/Operational
########## Windows Defender ##########
[WinEventLog://Microsoft-Windows-Windows Defender/Operational]
index = windows11
disabled = false
source = Microsoft-Windows-Windows Defender/Operational
blacklist = 1151,1150,2000,1002,1001,1000
# Additional Defender channel
[WinEventLog://Microsoft-Windows-Windows Defender/WHC]
index = windows11
disabled = false
########## PowerShell ##########
[WinEventLog://Microsoft-Windows-PowerShell/Operational]
index = windows11
disabled = false
source = Microsoft-Windows-PowerShell/Operational
blacklist = 4100,4105,4106,40961,40962,53504
########## PowerShell Core ##########
[WinEventLog://Microsoft-Windows-PowerShell/Core]
index = windows11
disabled = false
renderXml = true
########## Core Event Logs ##########
[WinEventLog://Application]
index = windows11
disabled = false
[WinEventLog://Security]
index = windows11
disabled = false
[WinEventLog://System]
index = windows11
disabled = false
########## RDP / Remote Access ##########
[WinEventLog://Microsoft-Windows-TerminalServices-LocalSessionManager/Operational]
index = windows11
disabled = false
[WinEventLog://Microsoft-Windows-TerminalServices-RemoteConnectionManager/Operational]
index = windows11
disabled = false
########## DNS Visibility ##########
[WinEventLog://Microsoft-Windows-DNS-Client/Operational]
index = windows11
disabled = false
renderXml = true
########## Task Scheduler (Persistence) ##########
[WinEventLog://Microsoft-Windows-TaskScheduler/Operational]
index = windows11
disabled = false
########## Firewall ##########
[WinEventLog://Microsoft-Windows-Windows Firewall With Advanced Security/Firewall]
index = windows11
disabled = false
########## BITS Client (Malware Downloader) ##########
[WinEventLog://Microsoft-Windows-Bits-Client/Operational]
index = windows11
disabled = false
########## Kernel Driver Loads ##########
[WinEventLog://Microsoft-Windows-Kernel-Driver/Operational]
index = windows11
disabled = false
########## AppLocker ##########
[WinEventLog://Microsoft-Windows-AppLocker/MSI and Script]
index = windows11
disabled = false
[WinEventLog://Microsoft-Windows-AppLocker/EXE and DLL]
index = windows11
disabled = false
```

Once the file is added under the **/local** folder, open **Services** as Administrator, locate the **Splunk Forwarder** service, open its properties, go to the **Log On** tab, select **Local System account**, apply the change, and restart the service.

Confirming Data Flow to Splunk

At this point, the Windows 11 VM should be forwarding its telemetry to Splunk.
In Splunk Web, go to **Search & Reporting** and query:

index=<your-index-name>



If you see incoming events, the forwarder is working correctly.

Installing n8n with Docker Compose

To run n8n, you'll use Docker Compose. Start by installing Docker and Docker Compose:

```
sudo apt install docker.io
```

```
leo@n8n:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 3 not upgraded.
Need to get 76.0 MB of archives.
After this operation, 288 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://pk.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://pk.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://pk.archive.ubuntu.com/ubuntu noble-updates/main amd64 runc amd64 1.3.3-0ubuntu1~24.04.2 [8,815 kB]
Get:4 http://pk.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.28-0ubuntu1~24.04.1 [38.4 MB]
Get:5 http://pk.archive.ubuntu.com/ubuntu noble-updates/main amd64 dns-root-data all 2024071801~ubuntu0.24.04.1 [5,918 B]
Get:6 http://pk.archive.ubuntu.com/ubuntu noble-updates/main amd64 dnsmasq-base amd64 2.90-2ubuntu0.1 [376 kB]
Get:7 http://pk.archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 28.2.2-0ubuntu1~24.04.1 [28.3 MB]
Get:8 http://pk.archive.ubuntu.com/ubuntu noble-updates/universe amd64 ubuntu-fan all 0.12.16+24.04.1 [34.2 kB]
Fetched 76.0 MB in 6min 14s (203 kB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 125904 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.8-1_amd64.deb ...
Unpacking pigz (2.8-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.7.1-1ubuntu2_amd64.deb ...
Unpacking bridge-utils (1.7.1-1ubuntu2) ...
Selecting previously unselected package runc.
Preparing to unpack .../2-runc_1.3.3-0ubuntu1~24.04.2_amd64.deb ...
Unpacking runc (1.3.3-0ubuntu1~24.04.2) ...
Selecting previously unselected package containerd.
Preparing to unpack .../3-containerd_1.7.28-0ubuntu1~24.04.1_amd64.deb ...
```

```
sudo apt install docker-compose
```

```
leo@n8n:~$ sudo apt  install docker-compose
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-compose python3-docker python3-dockerpty python3-docopt python3-dotenv python3-texttable python3-websocket
The following NEW packages will be installed:
  docker-compose python3-compose python3-docker python3-dockerpty python3-docopt python3-dotenv python3-texttable python3-websocket
0 upgraded, 8 newly installed, 0 to remove and 3 not upgraded.
Need to get 297 kB of archives.
After this operation, 1,589 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://pk.archive.ubuntu.com/ubuntu noble/universe amd64 python3-websocket all 1.7.0-1 [38.1 kB]
Get:2 http://pk.archive.ubuntu.com/ubuntu noble-updates/universe amd64 python3-docker all 5.0.3-1ubuntu1.1 [89.1 kB]
Get:3 http://pk.archive.ubuntu.com/ubuntu noble/universe amd64 python3-dockerpty all 0.4.1-5 [11.4 kB]
Get:4 http://pk.archive.ubuntu.com/ubuntu noble/universe amd64 python3-docopt all 0.6.2-6 [26.1 kB]
Get:5 http://pk.archive.ubuntu.com/ubuntu noble/universe amd64 python3-dotenv all 1.0.1-1 [22.3 kB]
Get:6 http://pk.archive.ubuntu.com/ubuntu noble/universe amd64 python3-texttable all 1.6.7-1 [11.0 kB]
Get:7 http://pk.archive.ubuntu.com/ubuntu noble/universe amd64 python3-compose all 1.29.2-6ubuntu1 [84.6 kB]
Get:8 http://pk.archive.ubuntu.com/ubuntu noble/universe amd64 docker-compose all 1.29.2-6ubuntu1 [14.0 kB]
Fetched 297 kB in 4s (78.0 kB/s)
Selecting previously unselected package python3-websocket.
(Reading database ... 126257 files and directories currently installed.)
Preparing to unpack .../0-python3-websocket_1.7.0-1_all.deb ...
Unpacking python3-websocket (1.7.0-1) ...
Selecting previously unselected package python3-docker.
Preparing to unpack .../1-python3-docker_5.0.3-1ubuntu1.1_all.deb ...
Unpacking python3-docker (5.0.3-1ubuntu1.1) ...
Selecting previously unselected package python3-dockerpty.
Preparing to unpack .../2-python3-dockerpty_0.4.1-5_all.deb ...
Unpacking python3-dockerpty (0.4.1-5) ...
Selecting previously unselected package python3-docopt.
Preparing to unpack .../3-python3-docopt_0.6.2-6_all.deb ...
Unpacking python3-docopt (0.6.2-6) ...
Selecting previously unselected package python3-dotenv.
Preparing to unpack .../4-python3-dotenv_1.0.1-1_all.deb ...
```

Create a directory for the n8n setup:

```
mkdir docker-compose ; cd n8n-compose
```

Create the Docker Compose configuration file:

```
sudo nano docker-compose.yaml
```

Paste the following into your **`docker-compose.yaml`** file:

```yaml
version: "3.9"

services:
  n8n:
    image: docker.n8n.io/n8nio/n8n:1.108.2
    container_name: n8n
    restart: always
    ports:
      - "5678:5678"
    environment:
      - N8N_HOST=0.0.0.0
      - N8N_PORT=5678
      - N8N_PROTOCOL=http
      - N8N_SECURE_COOKIE=false
      - GENERIC_TIMEZONE=America/Toronto
      - N8N_ENFORCE_SETTINGS_FILE_PERMISSIONS=true
      - N8N_RUNNERS_ENABLED=true
      - DB_SQLITE_POOL_SIZE=5
      - N8N_BLOCK_ENV_ACCESS_IN_NODE=false
      - N8N_GIT_NODE_DISABLE_BARE_REPOS=true
    volumes:
      - ./n8n_data:/home/node/.n8n
```

Save the file and exit by pressing Ctrl + X, then Y, and hit Enter.

Pull the image and start the container:

```
Sudo docker-compose  pull
```

```
Sudo docker-compose  up -d
```

```
ll
```

n8n needs correct ownership on the n8n_data directory. Update the permissions: `sudo chown -R 1000:1000 n8n_data/ && ll`

Restart n8n so it applies correctly:

```
Sudo docker-compose down
```

```
Sudo docker-compose up -d
```

```
Sudo docker ps
```

```
leo@n8n:~$
leo@n8n:~$ mkdir docker-compose ; cd docker-compose
leo@n8n:~/docker-compose$ sudo nano docker-compose.yaml
leo@n8n:~/docker-compose$ sudo docker-compose pull
Pulling n8n ... done
leo@n8n:~/docker-compose$ sudo docker-compose up -d
[sudo] password for leo:
Creating network "docker-compose_default" with the default driver
Creating n8n ... done
leo@n8n:~/docker-compose$ ls
docker-compose.yaml  n8n_data
leo@n8n:~/docker-compose$ ll
total 16
drwxrwxr-x 3 leo  leo  4096 Nov 24 10:20 ./
drwxr-x--- 6 leo  leo  4096 Nov 24 09:58 ../
-rw-r--r-- 1 root root  564 Nov 24 09:59 docker-compose.yaml
drwxr-xr-x 2 root root 4096 Nov 24 10:20 n8n_data/
leo@n8n:~/docker-compose$ sudo chown -R 1000:1000 n8n_data/
leo@n8n:~/docker-compose$ ll
total 16
drwxrwxr-x 3 leo  leo  4096 Nov 24 10:20 ./
drwxr-x--- 6 leo  leo  4096 Nov 24 09:58 ../
-rw-r--r-- 1 root root  564 Nov 24 09:59 docker-compose.yaml
drwxr-xr-x 2 leo  leo  4096 Nov 24 10:20 n8n_data/
leo@n8n:~/docker-compose$ sudo docker-compose down
Stopping n8n ... done
Removing n8n ... done
Removing network docker-compose_default
leo@n8n:~/docker-compose$ sudo docker-compose up -d
Creating network "docker-compose_default" with the default driver
Creating n8n ... done
leo@n8n:~/docker-compose$ docker ps
```

Now open your browser on the host machine and go to: **`http://{your-n8n-vm-ip}:5678`**

You'll see the n8n signup page. Complete the form and continue

Generating Test Event Logs (Failed Logons)

To generate failed login attempts, try connecting to the Windows 11 VM using an incorrect username or password (via RDP or SSH). Windows will log these attempts using:

- **Event ID 4625** – Failed logon
- **Event ID 4624** – Successful logon

Creating a Failed Logon Alert in Splunk

Return to Splunk and create a search that identifies failed logon attempts. Remember: Splunk search is case-sensitive

`(EventCode ≠ eventcode)`.

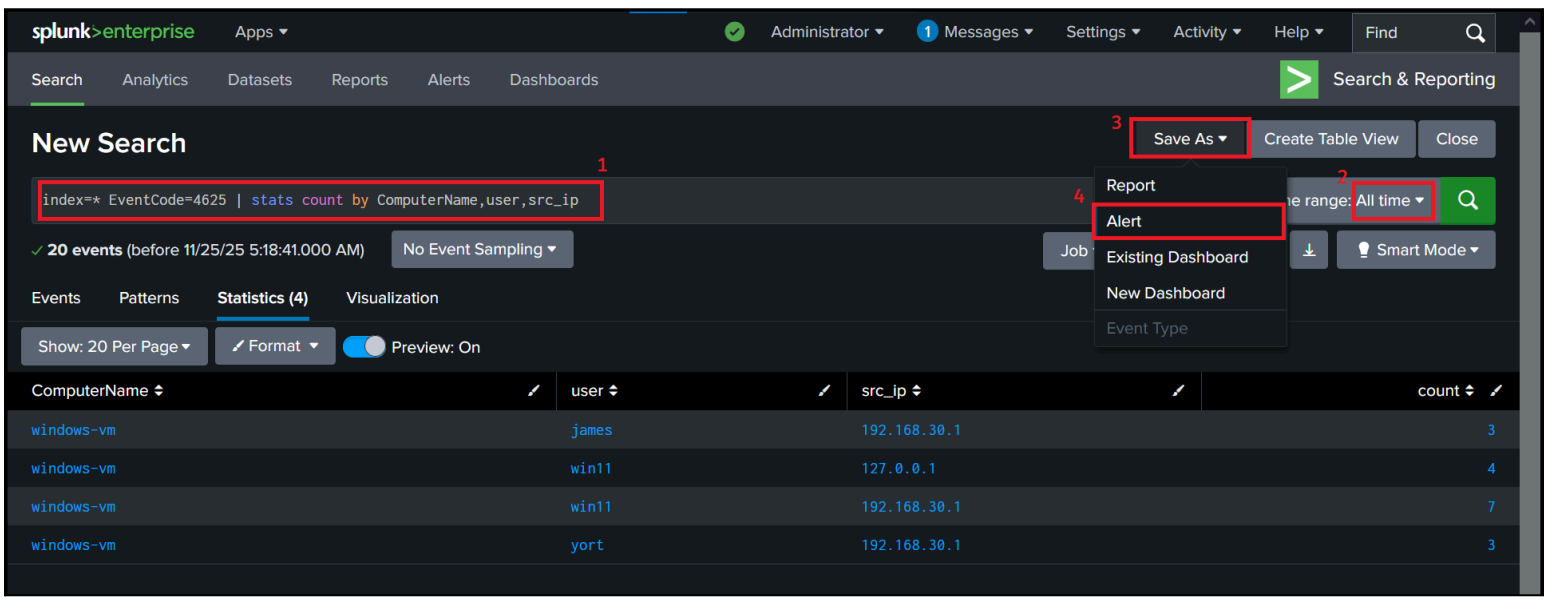
Run:

```
index=* EventCode=4625 | stats count by ComputerName, user, src_ip
```

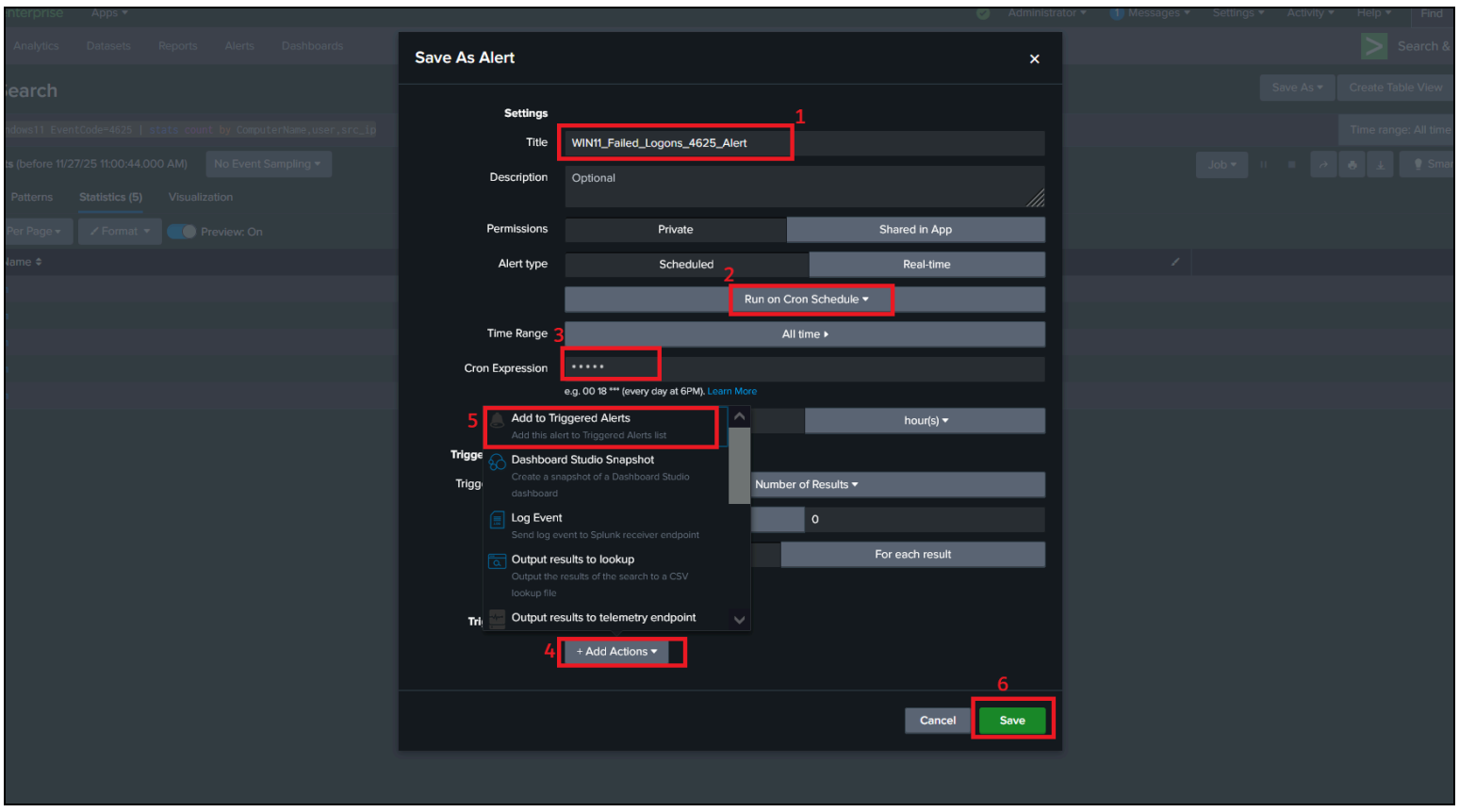
Set the time range to **All time**.

To save this as an alert:

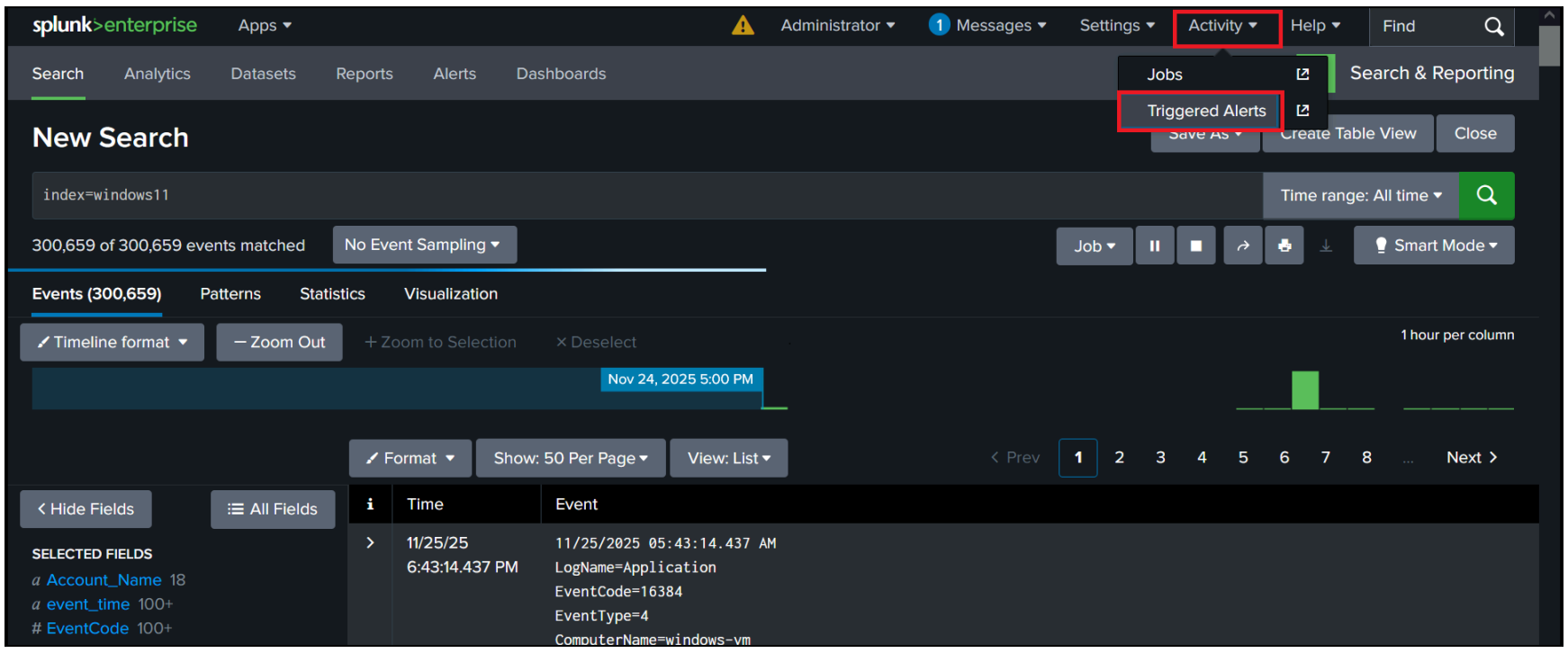
1. Click **Save As → Alert**



2. Give it a clear name such as: **WIN11_Failed_Logons_4625_Alert** and save the alert.



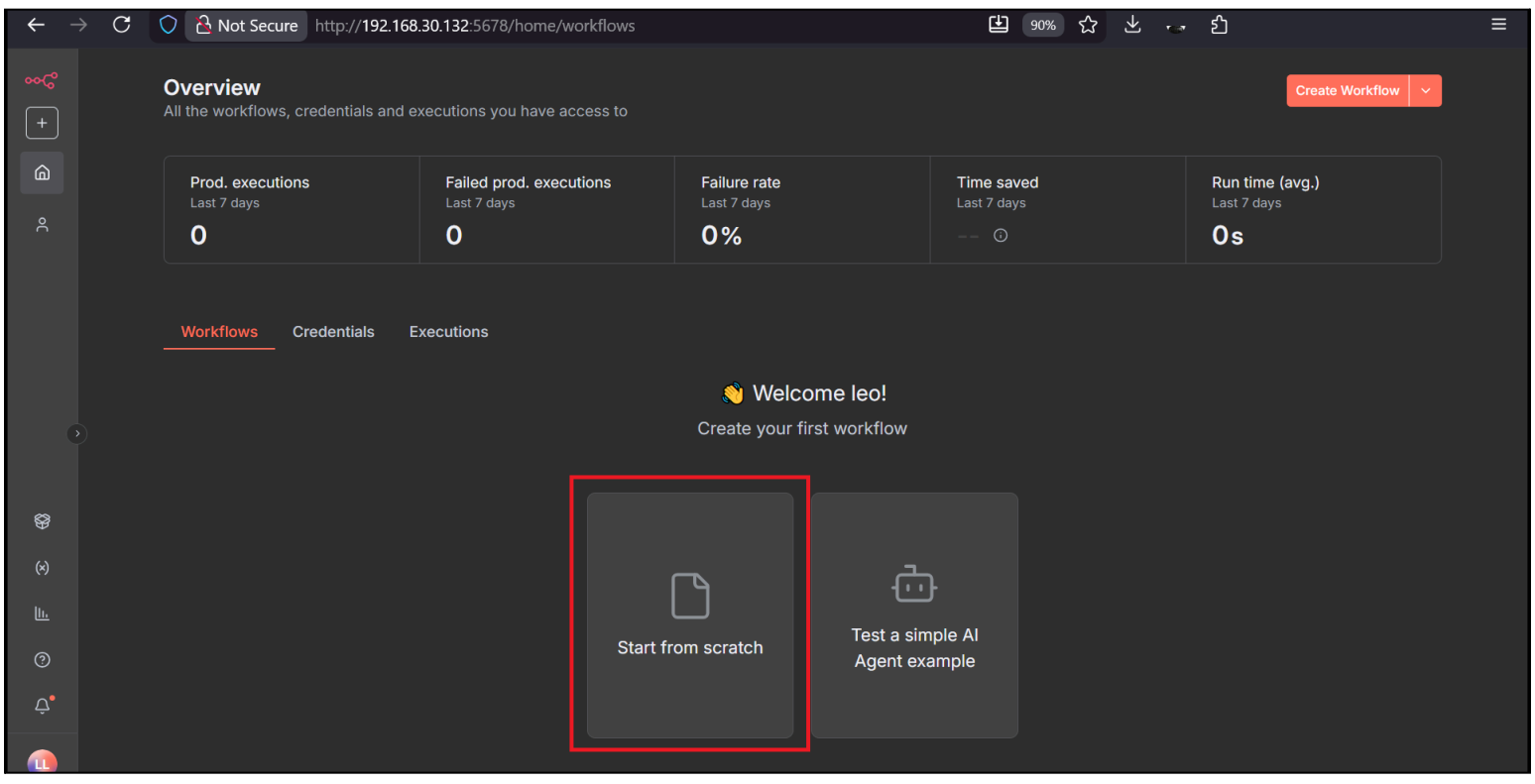
To view triggered alerts, go to **Activity** → **Triggered Alerts**.



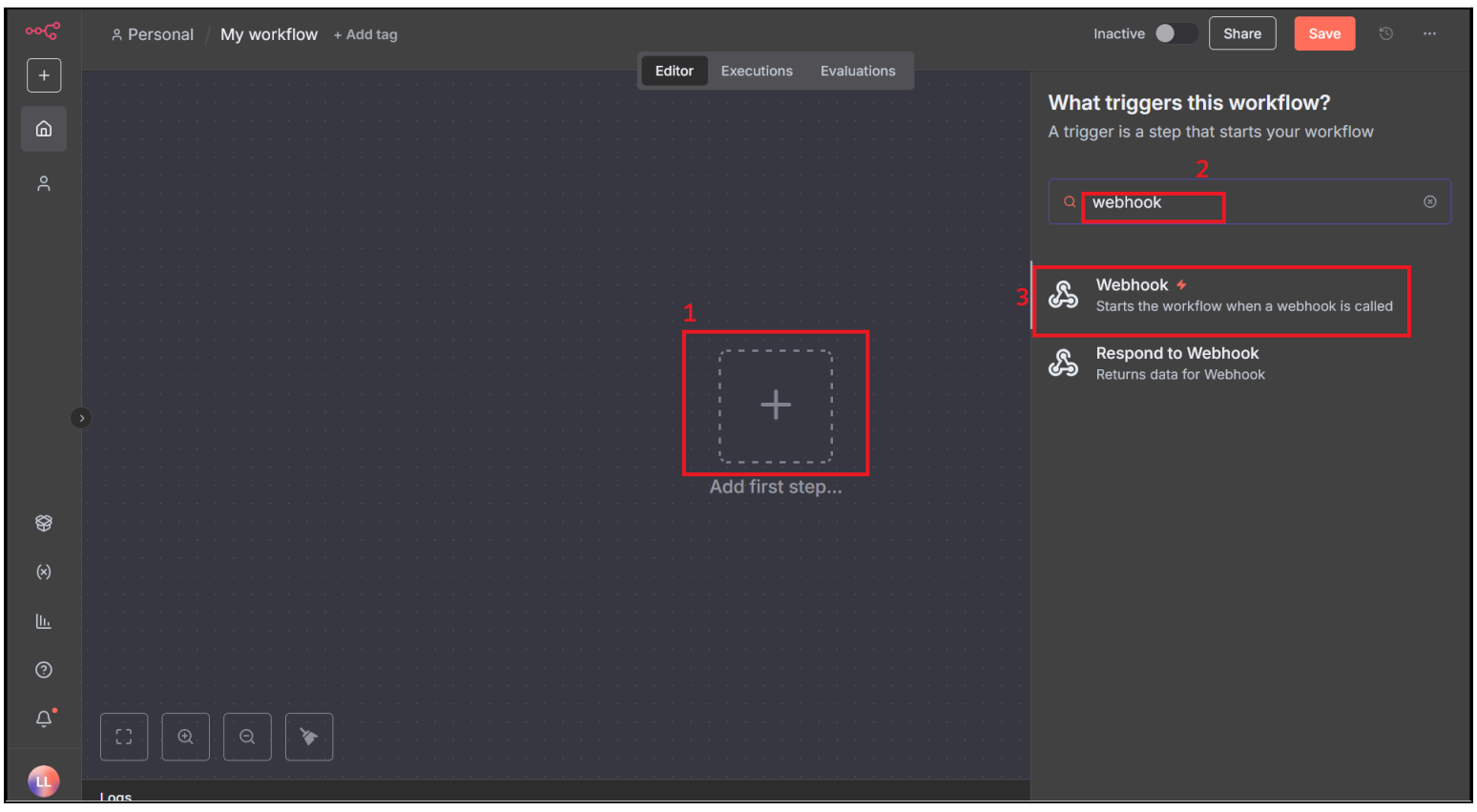
Setting Up the Webhook in n8n

In n8n:

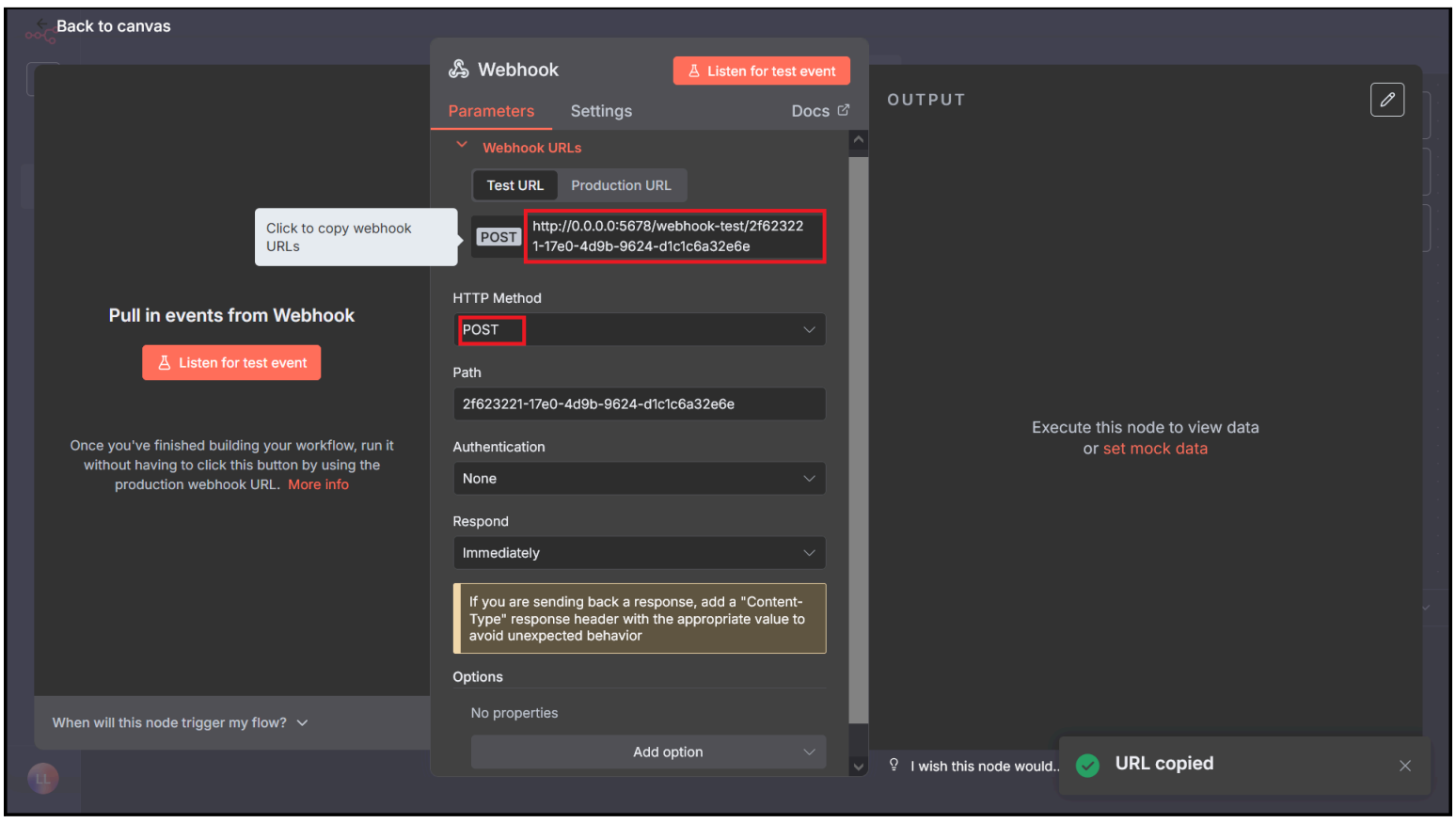
1. Create a new workflow.
2. Click **Start from Scratch**.



3. Click **+** and search for **Webhook,** then select it

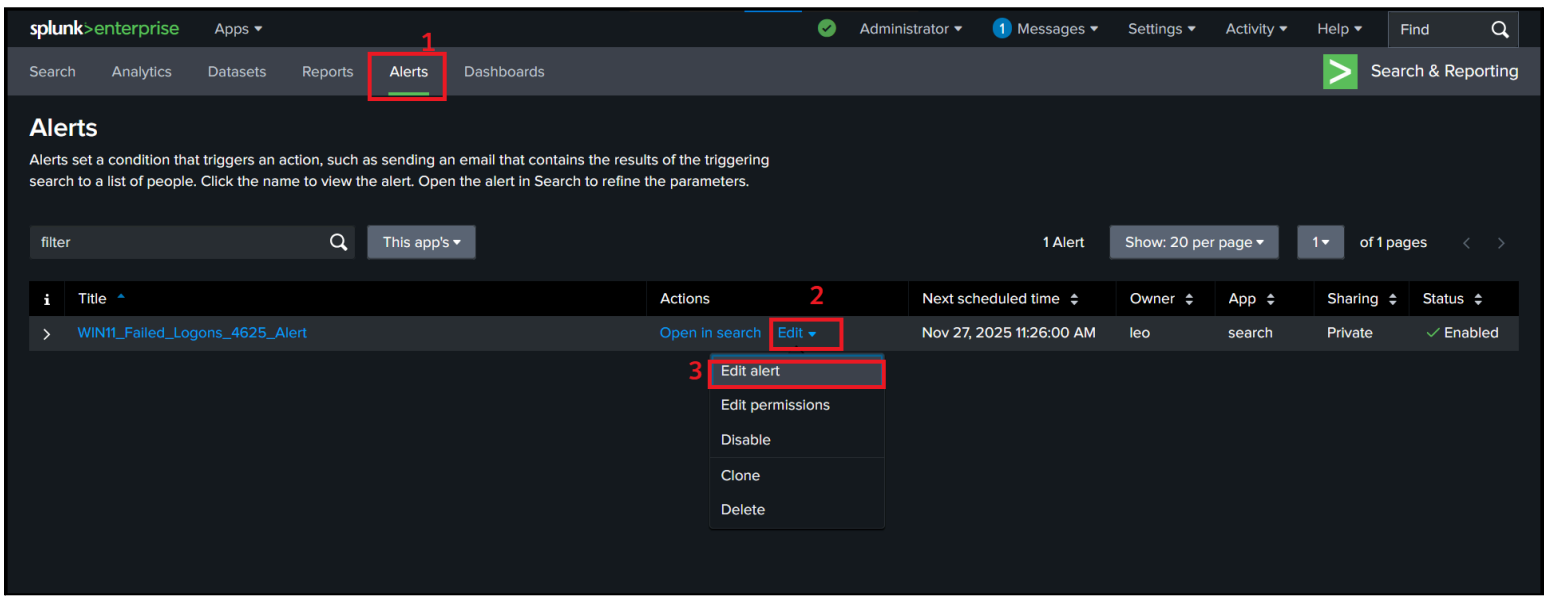


4. Set the HTTP method to **POST** and copy the generated webhook URL.

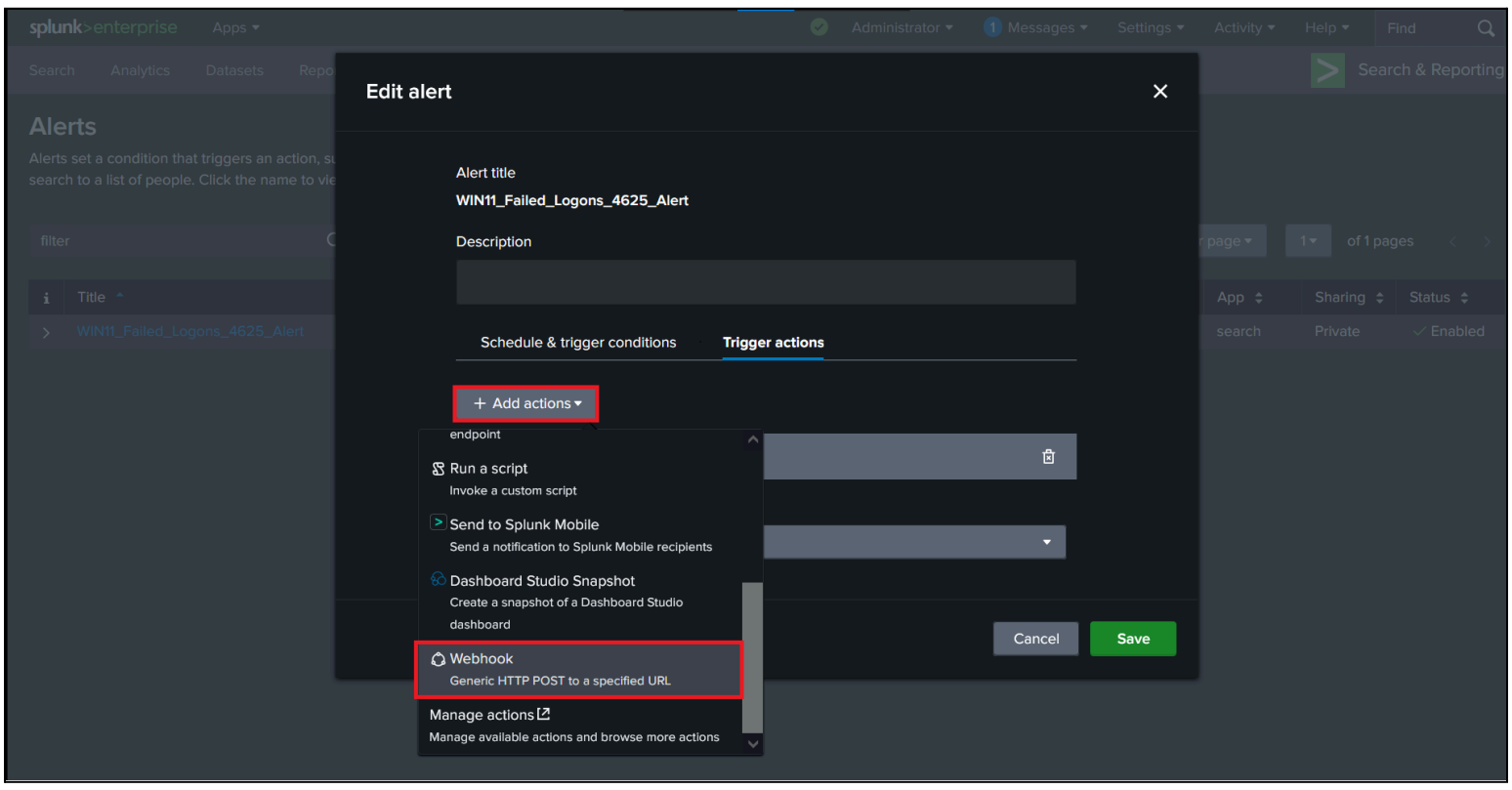


Now, add this URL to your Splunk alert:

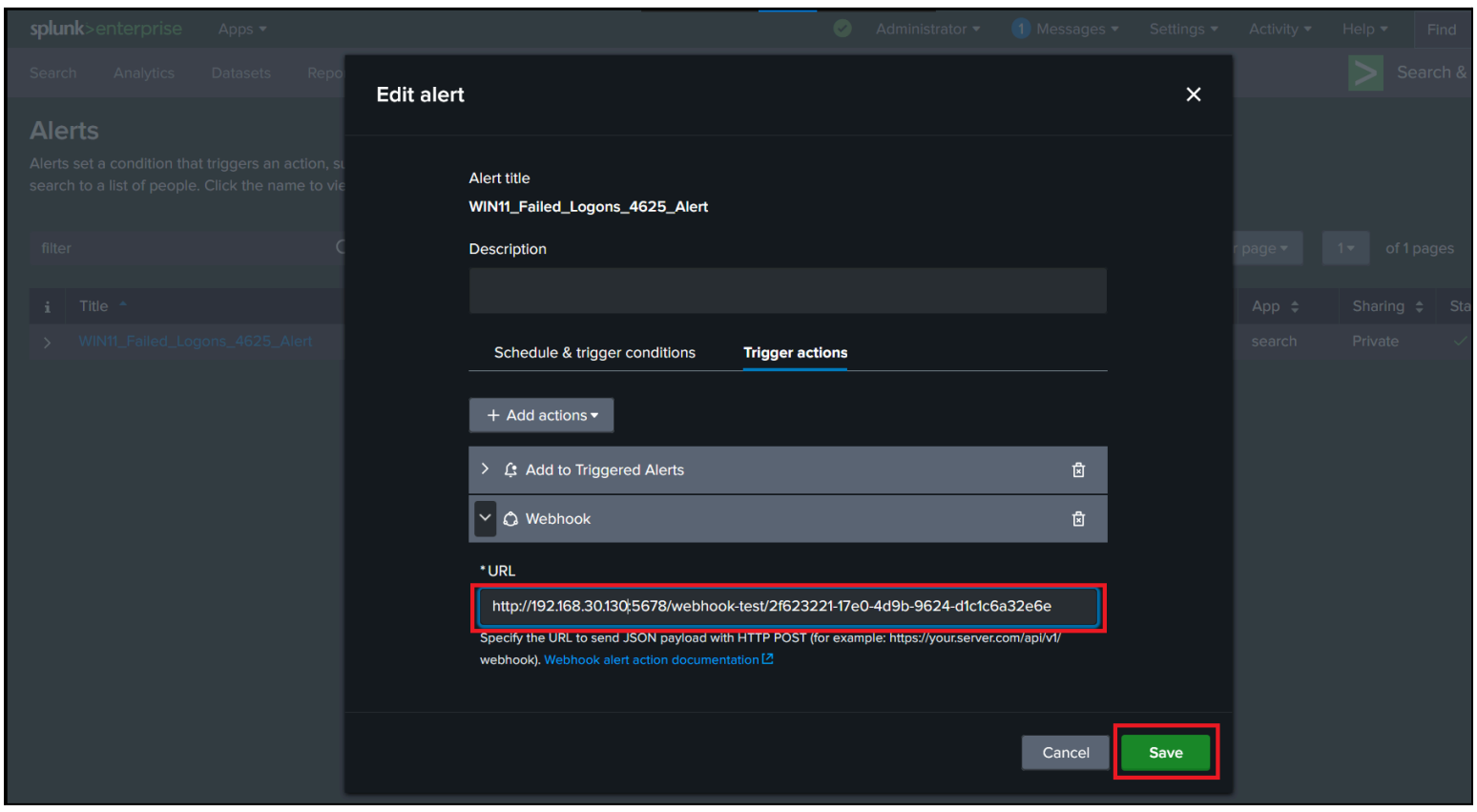
1. Go to **Search & Reporting**
2. Click **Alerts**
3. Find your alert, click **Edit → Edit Alert**



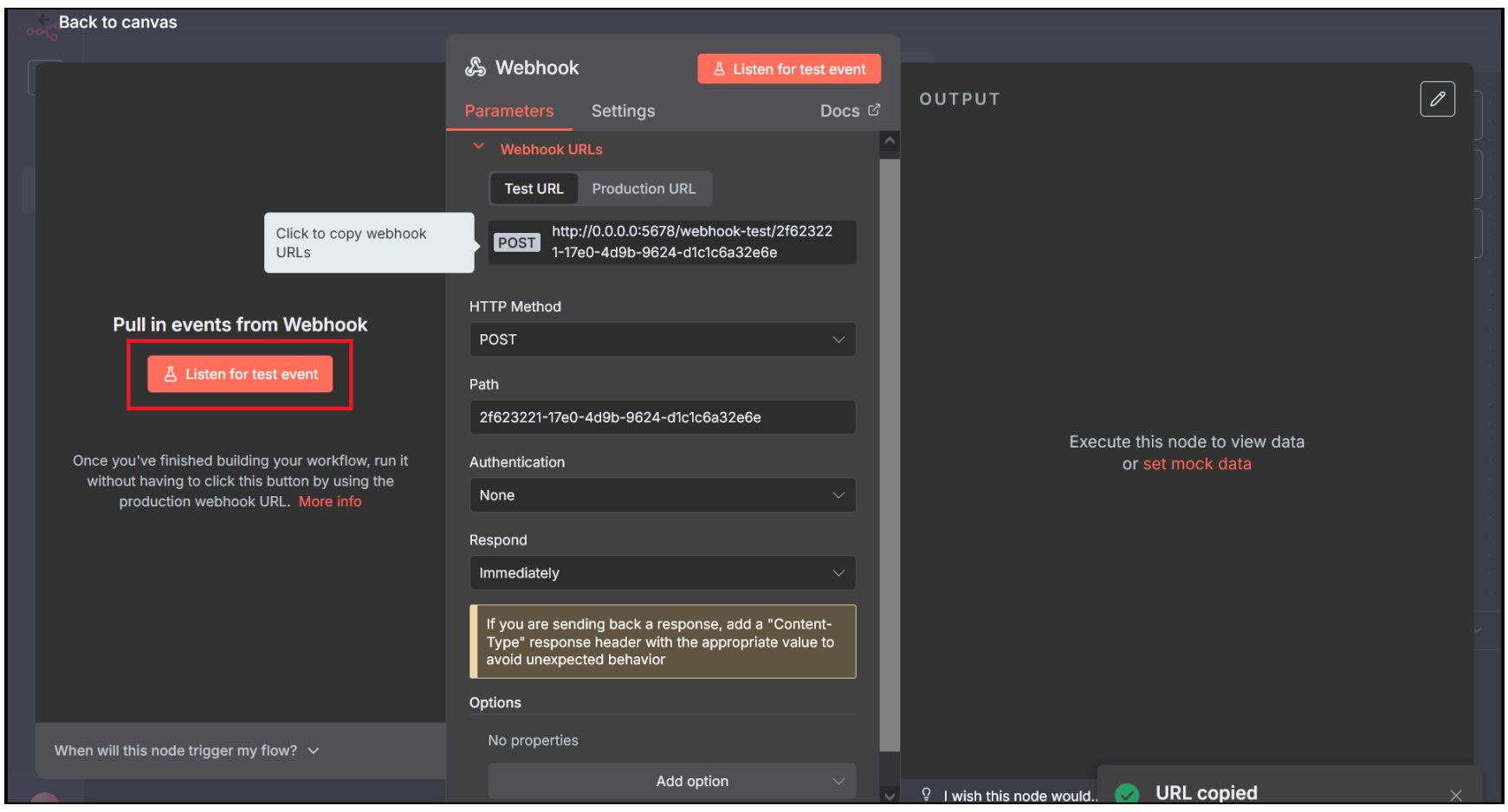
4. Under **Add Actions**, select **Webhook**



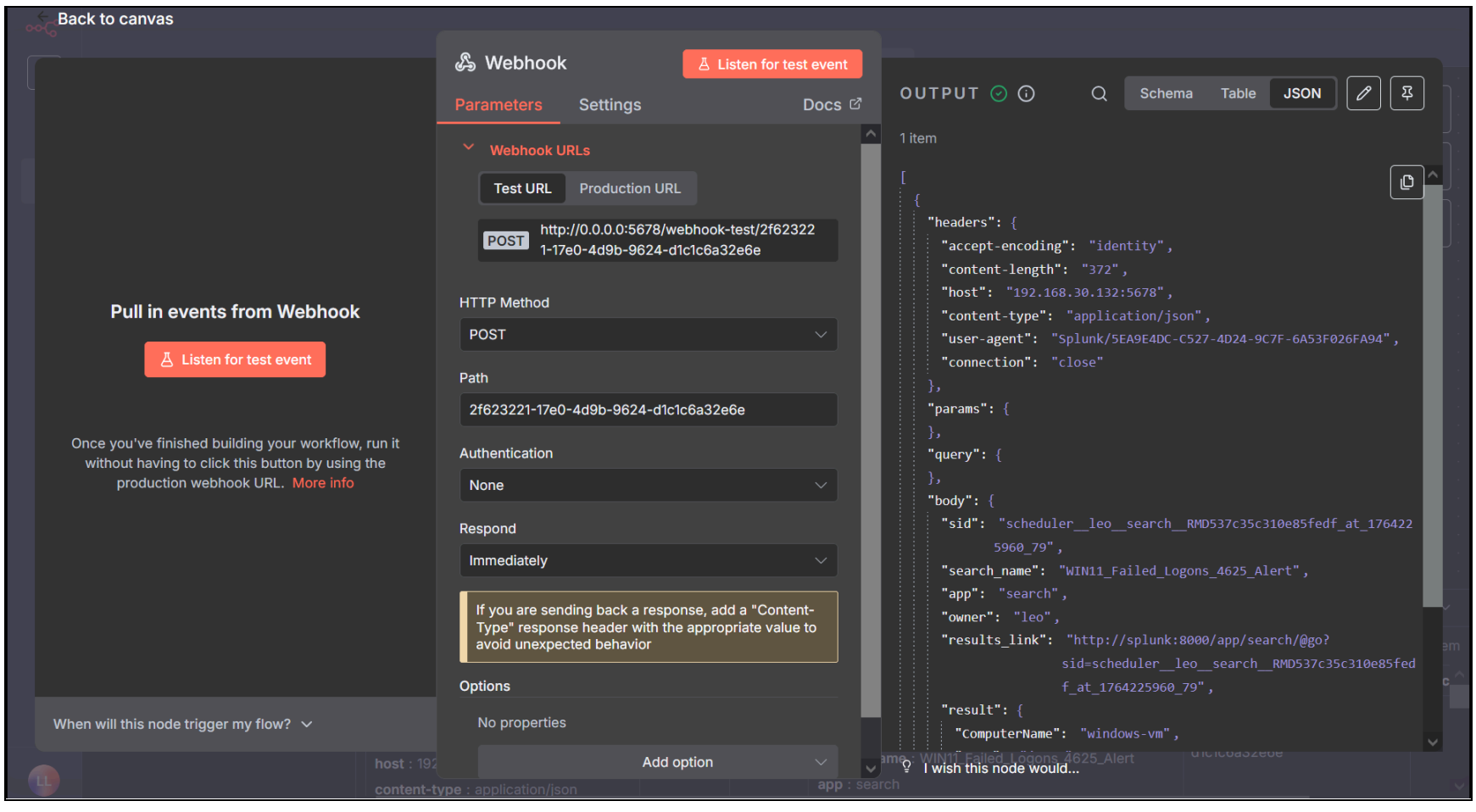
5. Paste the Webhook URL, replace **0.0.0.0** with your **Splunk VM IP**, then save it..



Return to n8n and click **Listen for Test Events**.

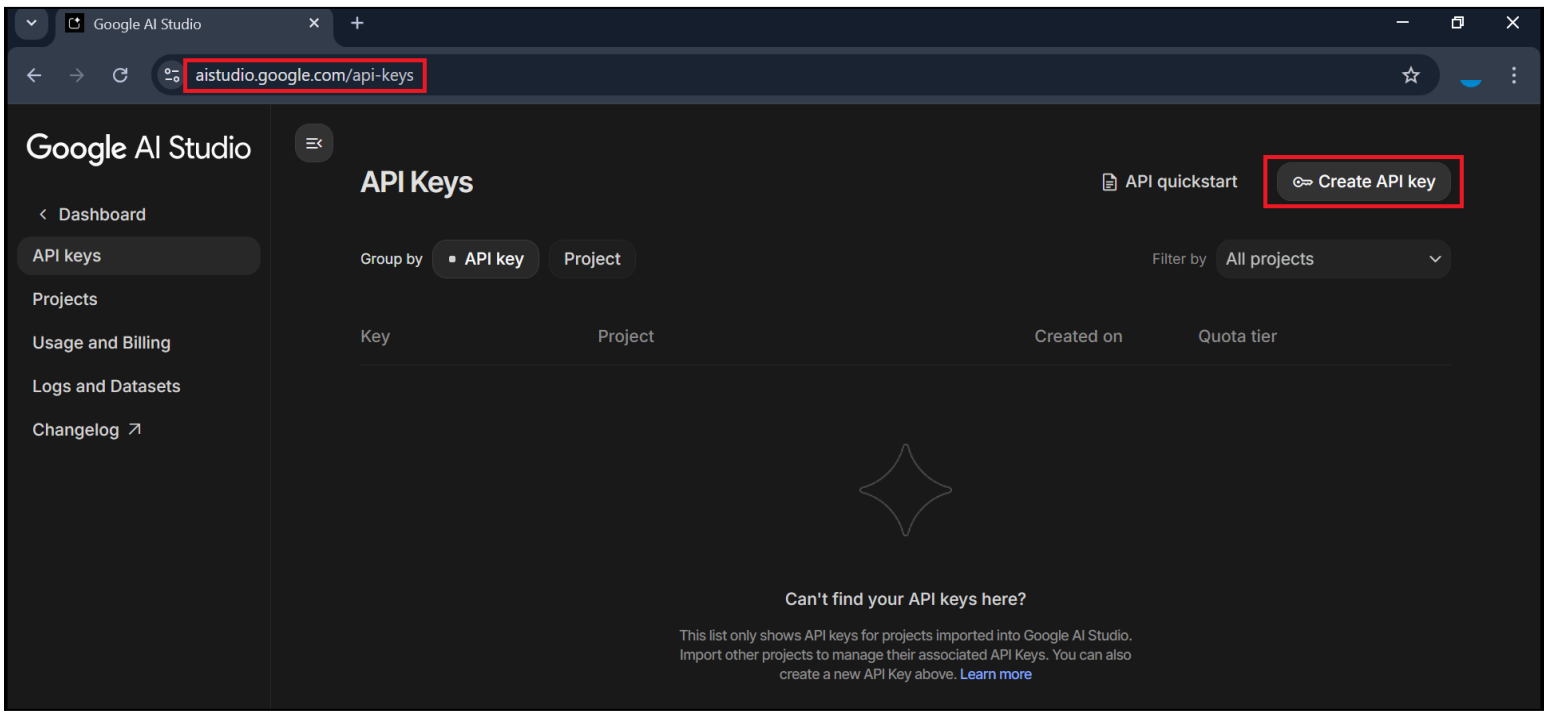


When the alert triggers (your Splunk alert runs every minute), n8n will capture the incoming data. Pin the Webhook node so it stays active.

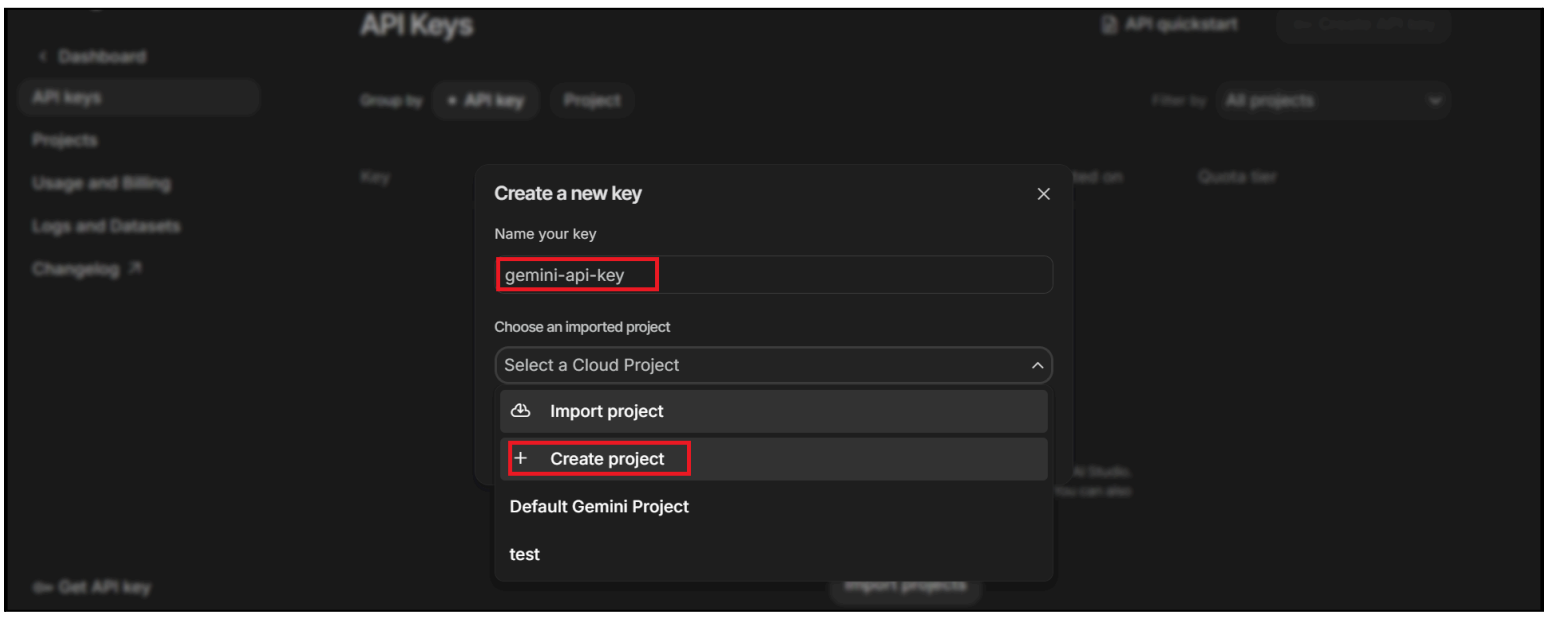


Getting Your Gemini API Key

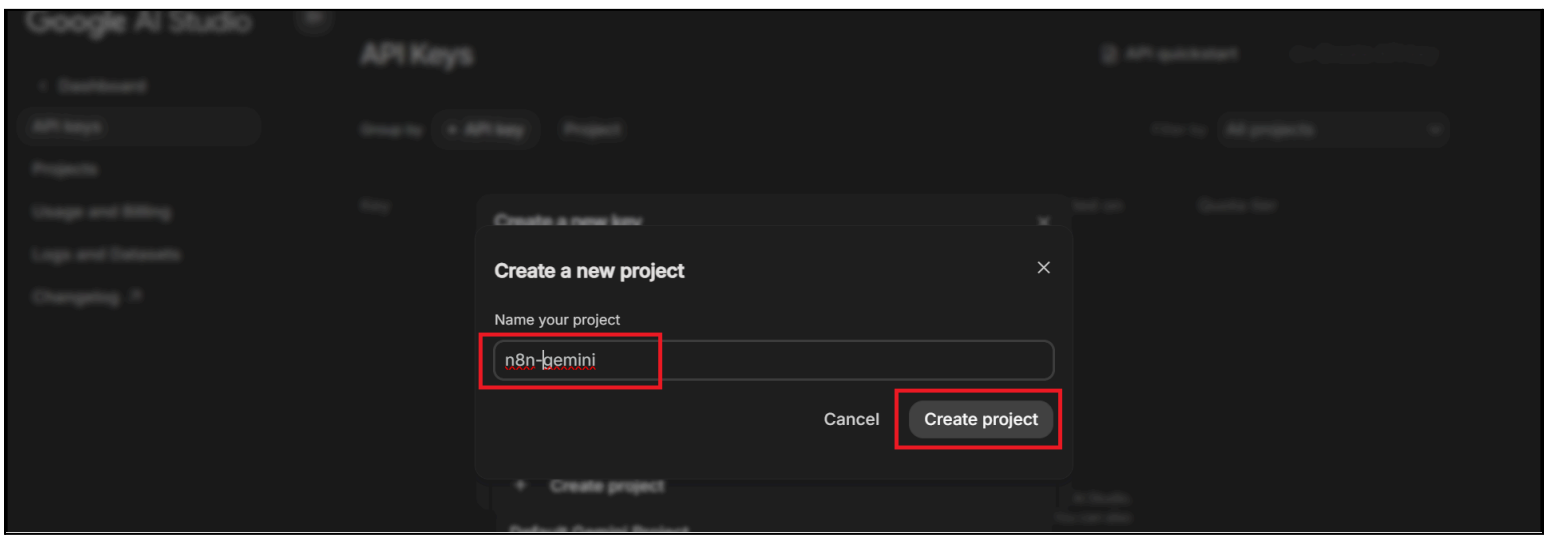
1. Go to aistudio.google.com
2. Click **Create API Key** (top-right corner)



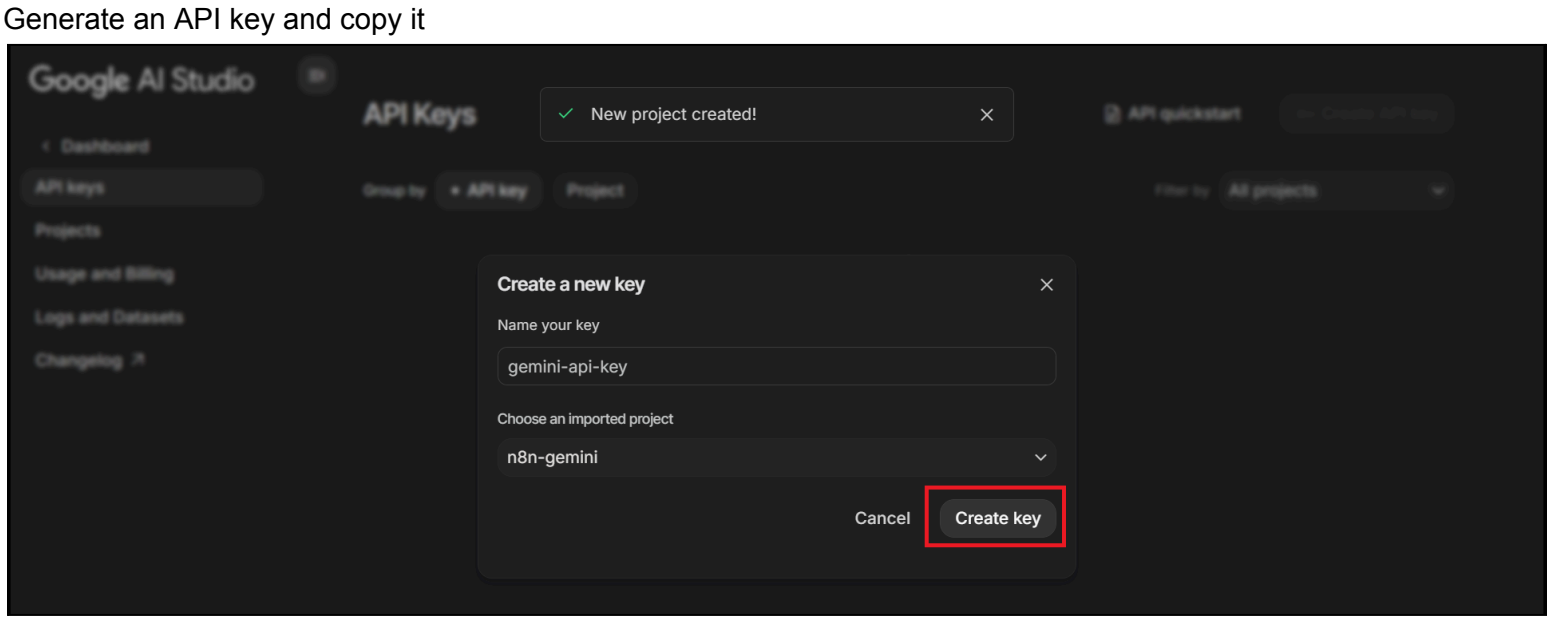
3. Name your key, and under “Select a Cloud Project,” click **Create Project**.



4. Name your project and create it



5. Generate an API key and copy it

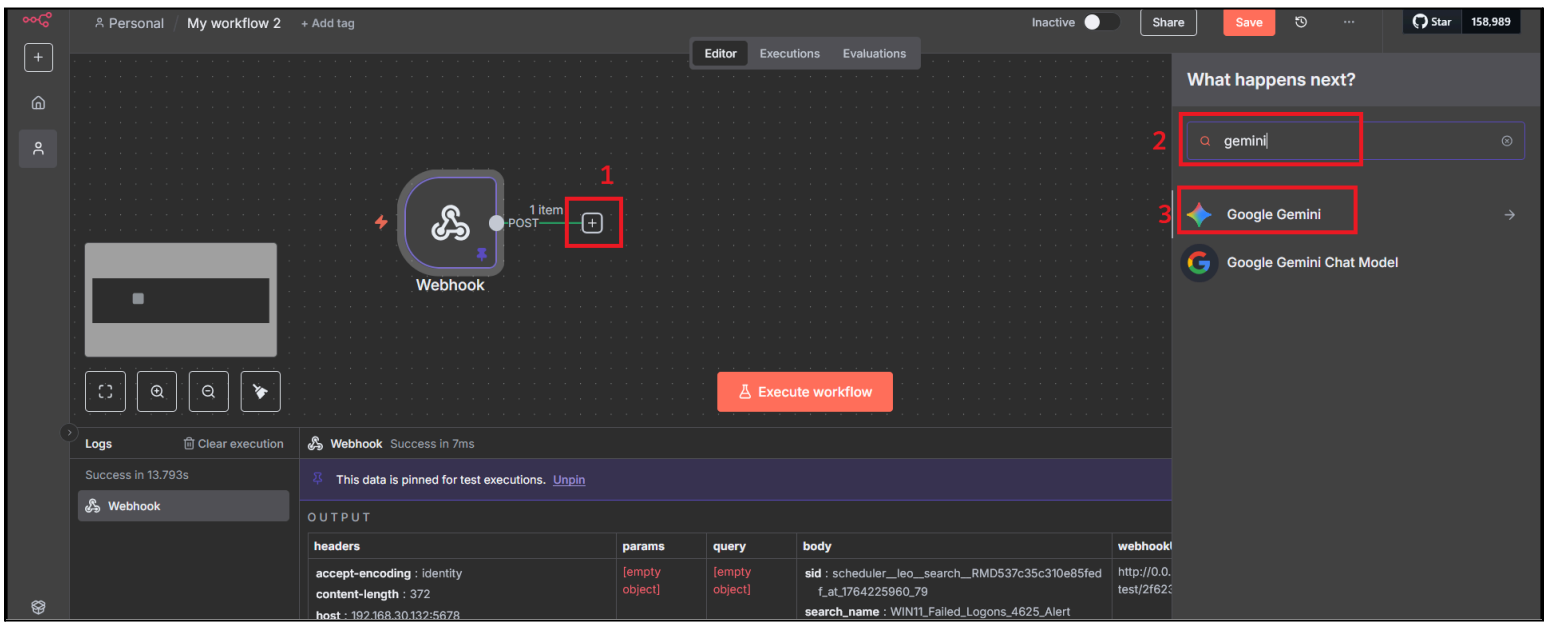


You can now use this API key in the Gemini node in n8n.

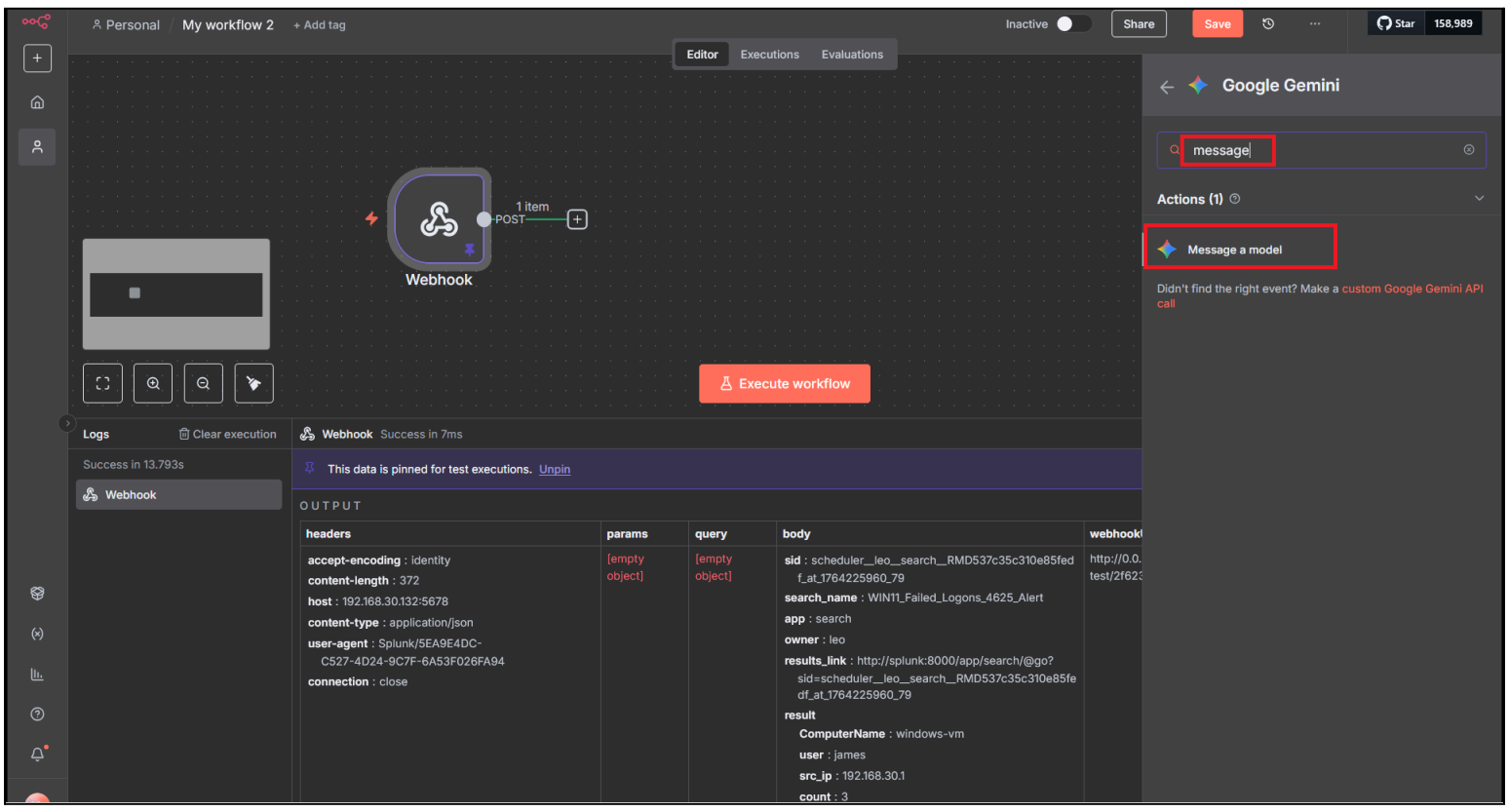
Adding Gemini to the Workflow

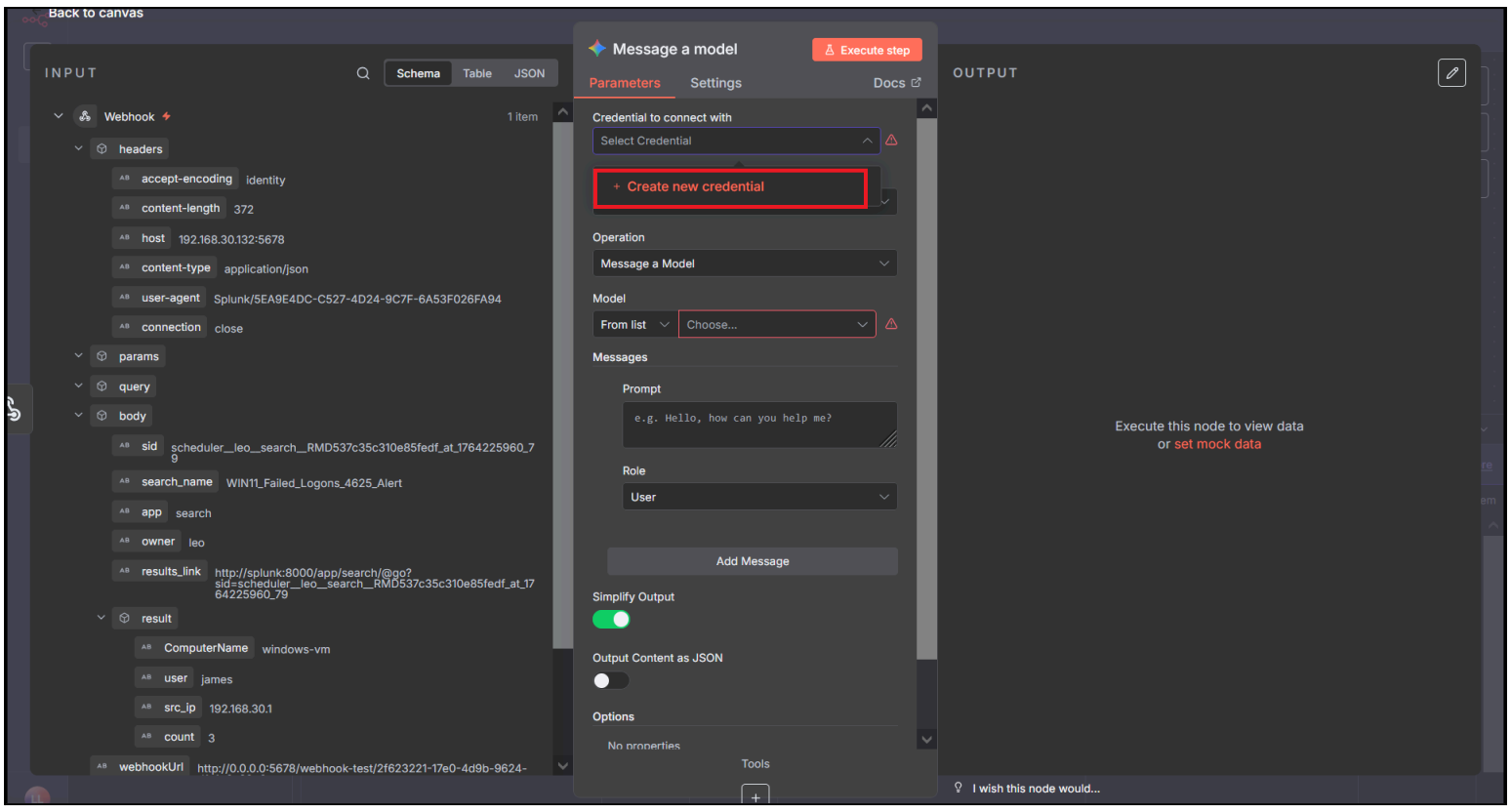
To add Gemini integration:

1. Click **+** Search for **Gemini**

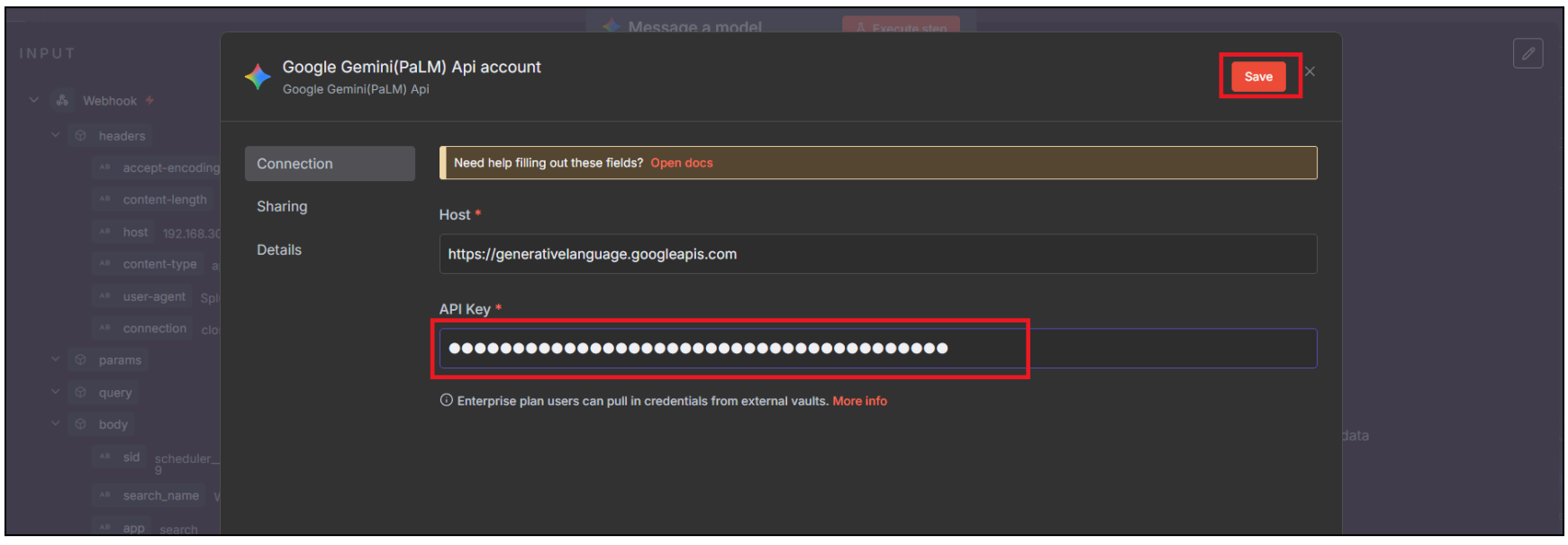


2. Select **Message a Model**

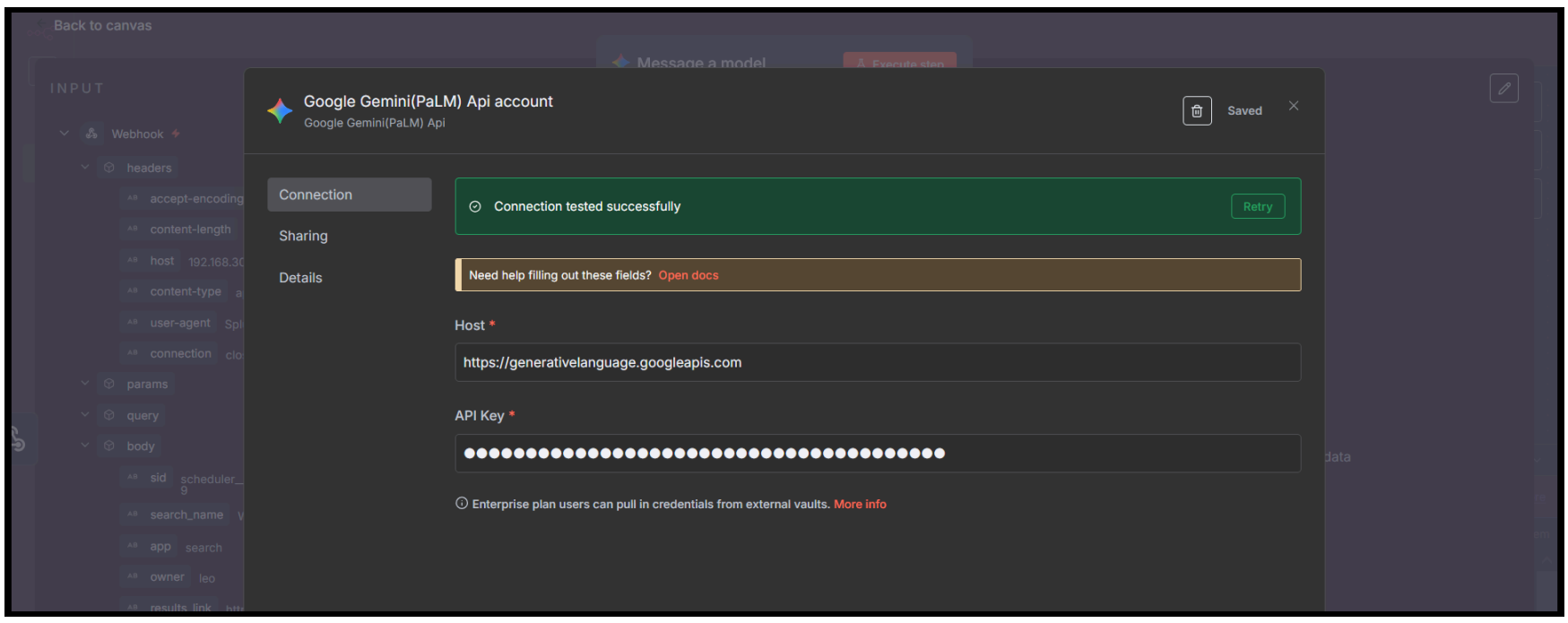




Enter your Gemini API key and click **Save**.



You should see the message **connection tested successfully**.



Add the Main Triage Prompt

Under **Prompt**, paste the following:

Message 1 (Model role):

```
Role:
You are a Tier 1 SOC Analyst assistant responsible for analyzing alerts and producing structured triage output.

Objective:
Given any alert payload (logs, metadata, IOCs, authentication events, network indicators, or host telemetry), produce a Tier-1 triage package.

Tasks:
1. Summary
  - Describe what triggered the alert.
  - Identify affected assets: users, hosts, IPs, applications.
  - Classify the activity (for example: suspicious login, malware detection, anomalous behavior).

2. IOC Enrichment
  - Evaluate indicators (IP, domain, URL, file hash) as malicious, suspicious, or benign.
  - Include known threat actors, malware families, campaigns, and confidence levels where applicable.
  - For IPs, query AbuseIPDB and include Abuse Confidence Score, report count, last report date, and categories. If the lookup fails, state that explicitly.

3. MITRE Mapping and Severity
  - Map relevant MITRE ATT&CK tactics and techniques.
  - Assign a severity rating: Low, Medium, High, or Critical, with a brief justification.

4. Recommended Actions
  - Provide Tier-1 investigation steps.
  - Include escalation guidance for Tier-2.
  - Suggest containment actions such as blocking IPs, isolating hosts, or resetting credentials.

Output format:
Summary
IOC Enrichment
MITRE Mapping
Severity
Recommended Actions

Constraints:
- Keep the response concise and factual.
- Base all reasoning strictly on the alert content and enrichment results.
```

Set **Role** to **Model**.

Add the Formatting Prompt

Click **Add Message** and insert:

Message 2 (Model role):

```
Format the final response in a clean, consistent structure. Follow this layout exactly:

Summary - Provide a brief and clear explanation of the alert and the suspicious activity.

IOC Enrichment - For each indicator (IP, domain, hash, URL), include reputation details, threat associations, and confidence levels. For IPs, query AbuseIPDB and include the Abuse Confidence Score, report count, last report date, and categories. Clearly note if any lookup fails.
```

```
Severity Assessment - Provide a justified severity rating based on available evidence and the mapped MITRE ATT&CK techniques.

Recommended Actions - Include Tier-1 investigation steps, escalation conditions, and appropriate containment actions.

Constraints:
- Keep the output concise and operational.
- Do not invent or assume missing details; call out incomplete information.
- Base all analysis strictly on the provided alert data and enrichment results.
```

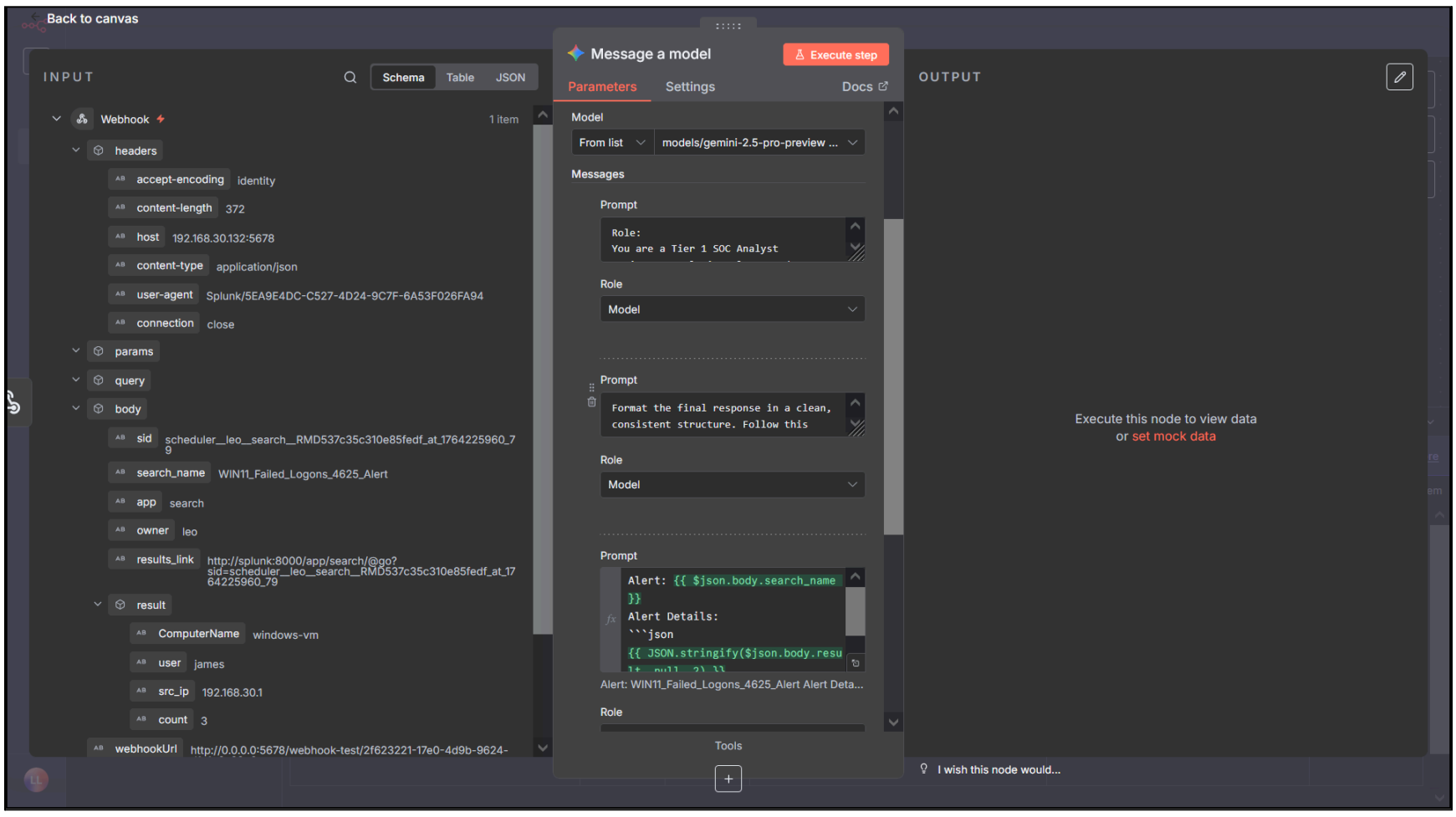
Set **Role** to **Model**.

Add the Alert Payload Prompt

Click **Add Message** again and paste:

Message 3 (Model role):

```
Alert: {{ $json.body.search_name }}
Alert Details:
```json
{{ JSON.stringify($json.body.result, null, 2) }}
```

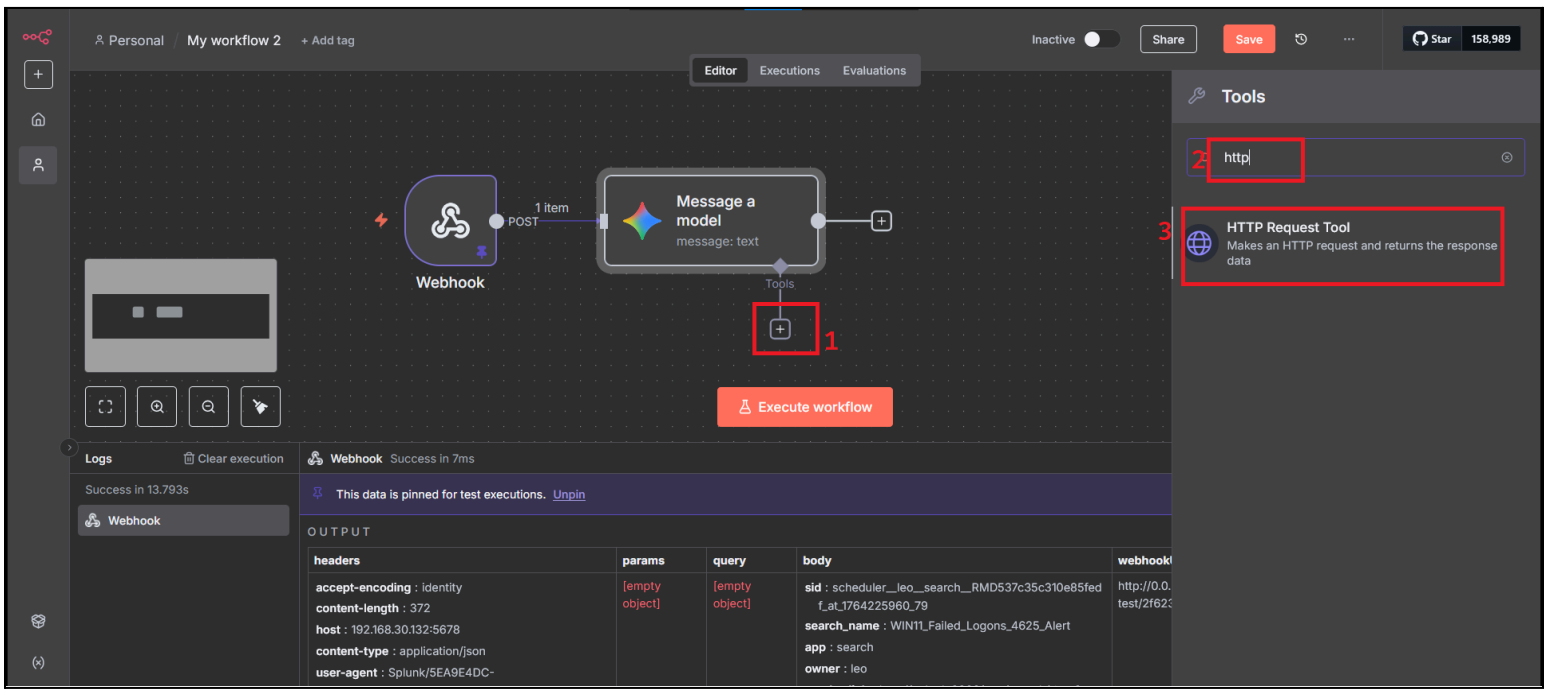


You can adjust these prompts anytime if you need tighter responses or more detail.

Add AbuseIPDB Integration

To integrate AbuseIPDB, you need to add an **HTTP Request** tool in n8n.

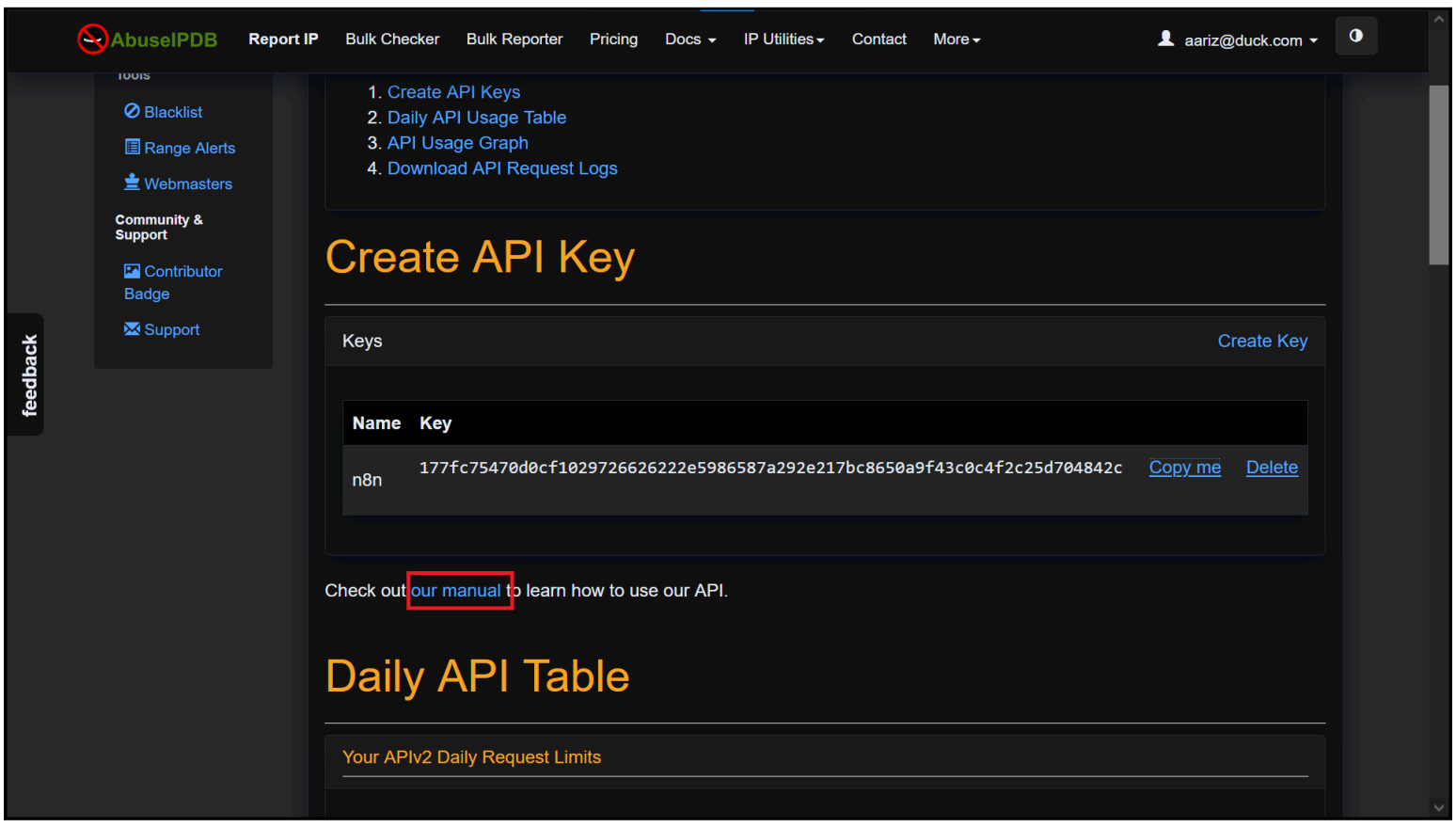
1. In your workflow, click the **+** icon under **Tools**.
2. Search **HTTP** and select **HTTP Request**.



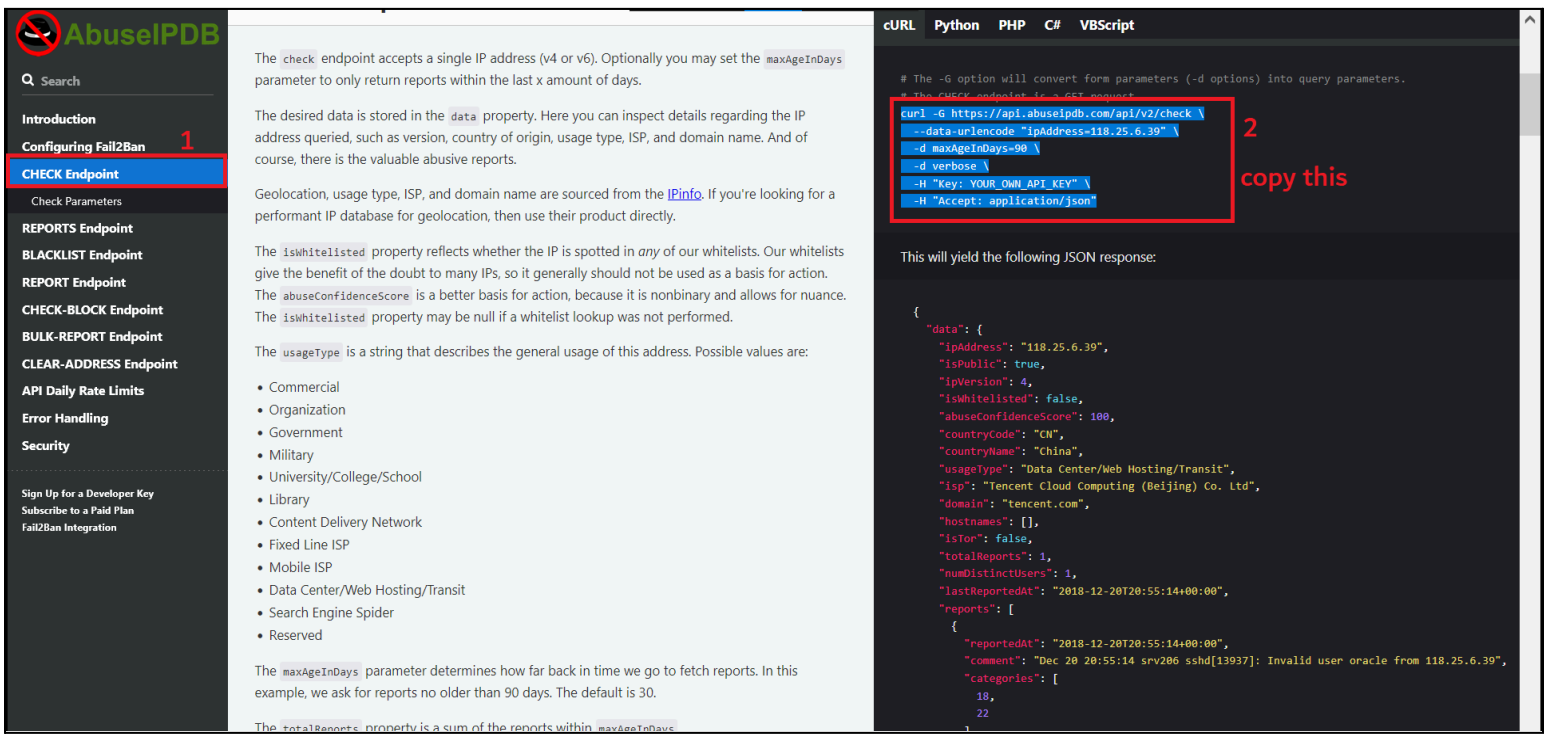
Next, create an AbuseIPDB account:

1. Go to **abuseipdb.com**, create an account, and open the **API** page.
```

2. Generate an API key.

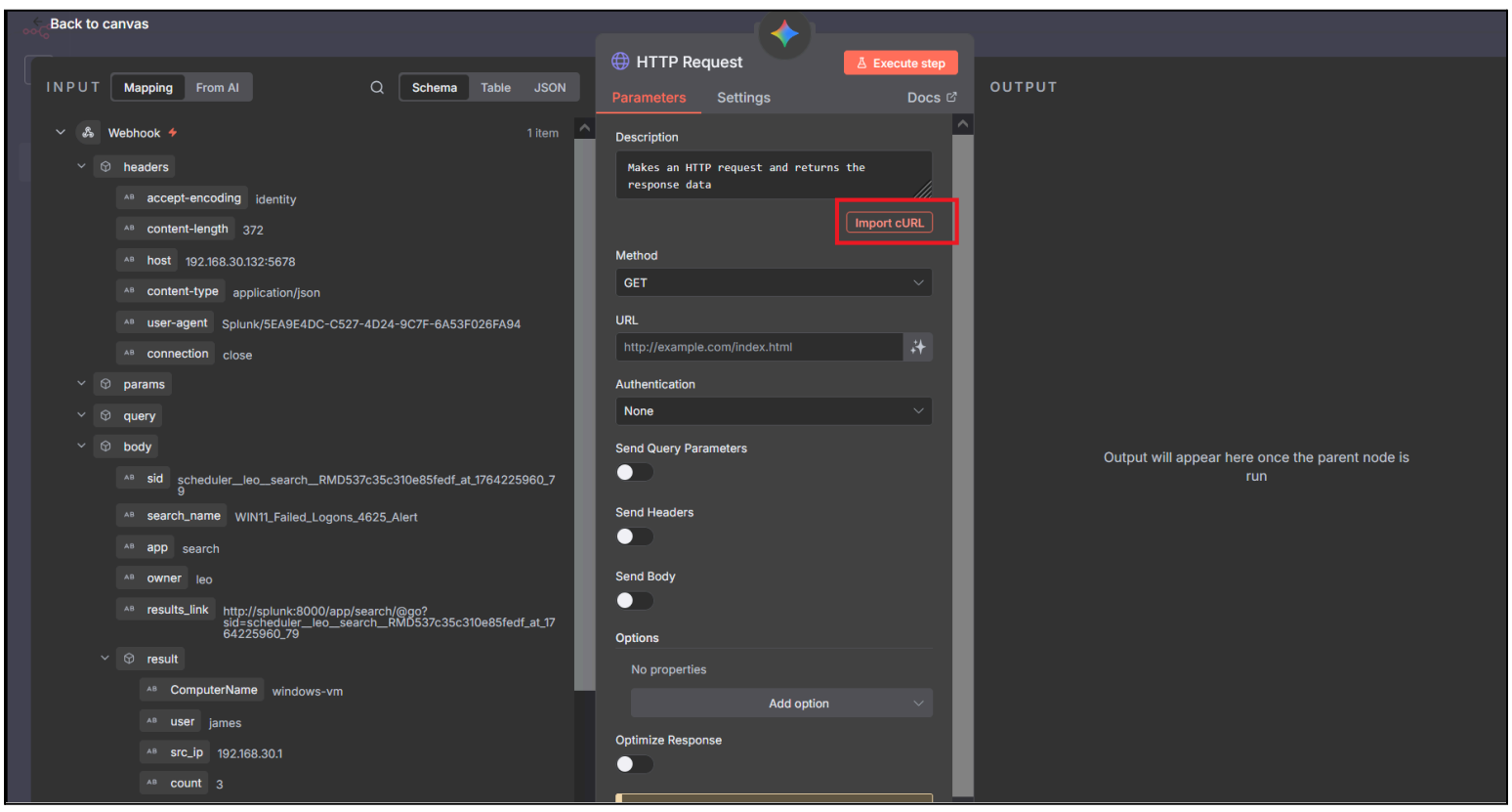


3. Under the **Check Endpoint**, copy the provided cURL command.

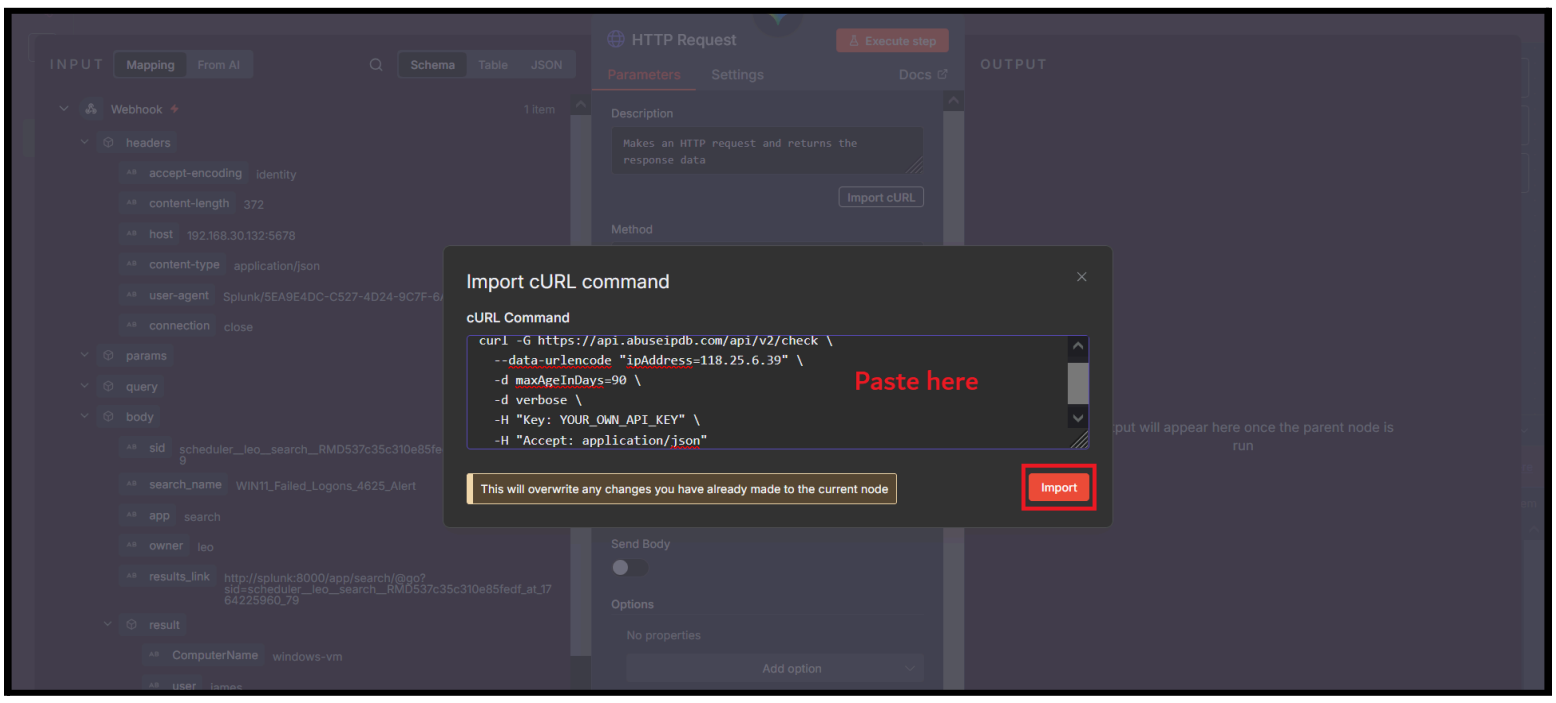


Back in n8n:

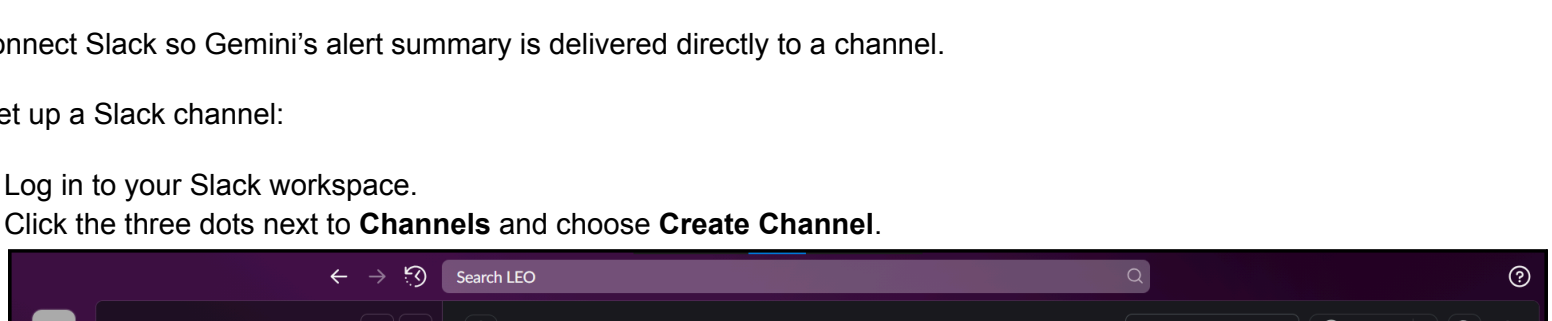
1. Click **Import cURL** in the HTTP Request node.



2. Paste the cURL command and click **Import**.



3. For the `ipAddress` parameter, click the model icon and select **Let the model define these parameters**.

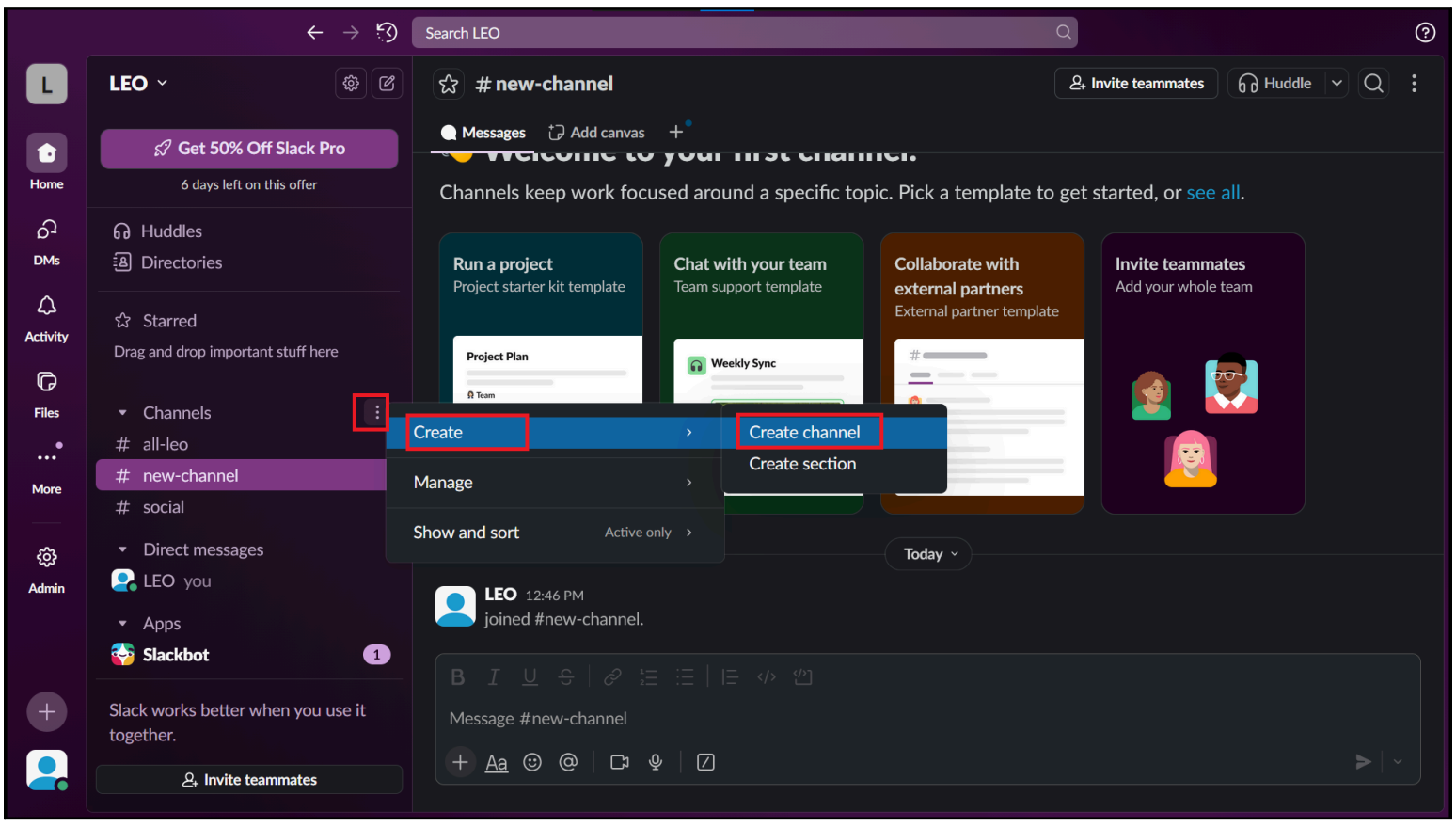


Add Slack Integration

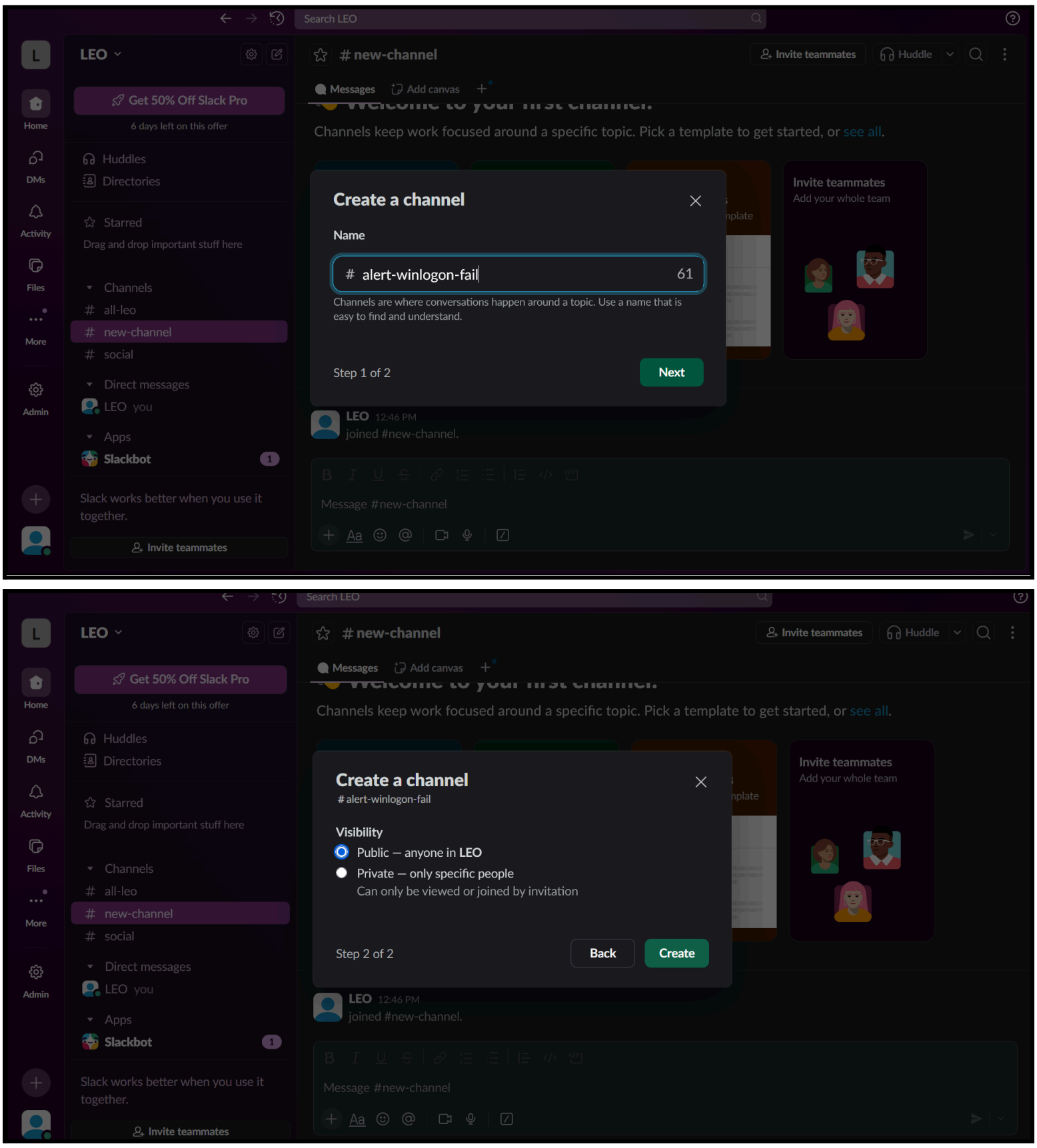
Now connect Slack so Gemini's alert summary is delivered directly to a channel.

First, set up a Slack channel:

1. Log in to your Slack workspace.
2. Click the three dots next to **Channels** and choose **Create Channel**.

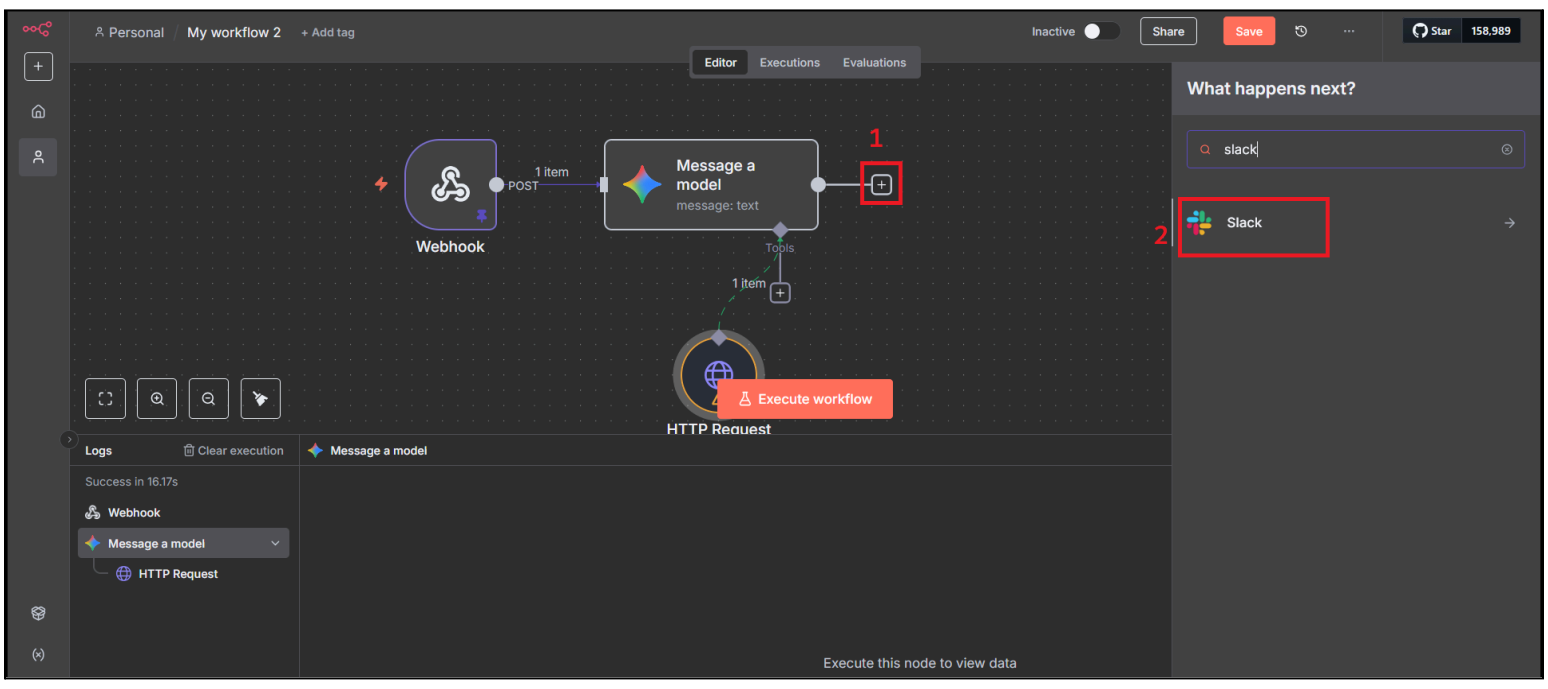


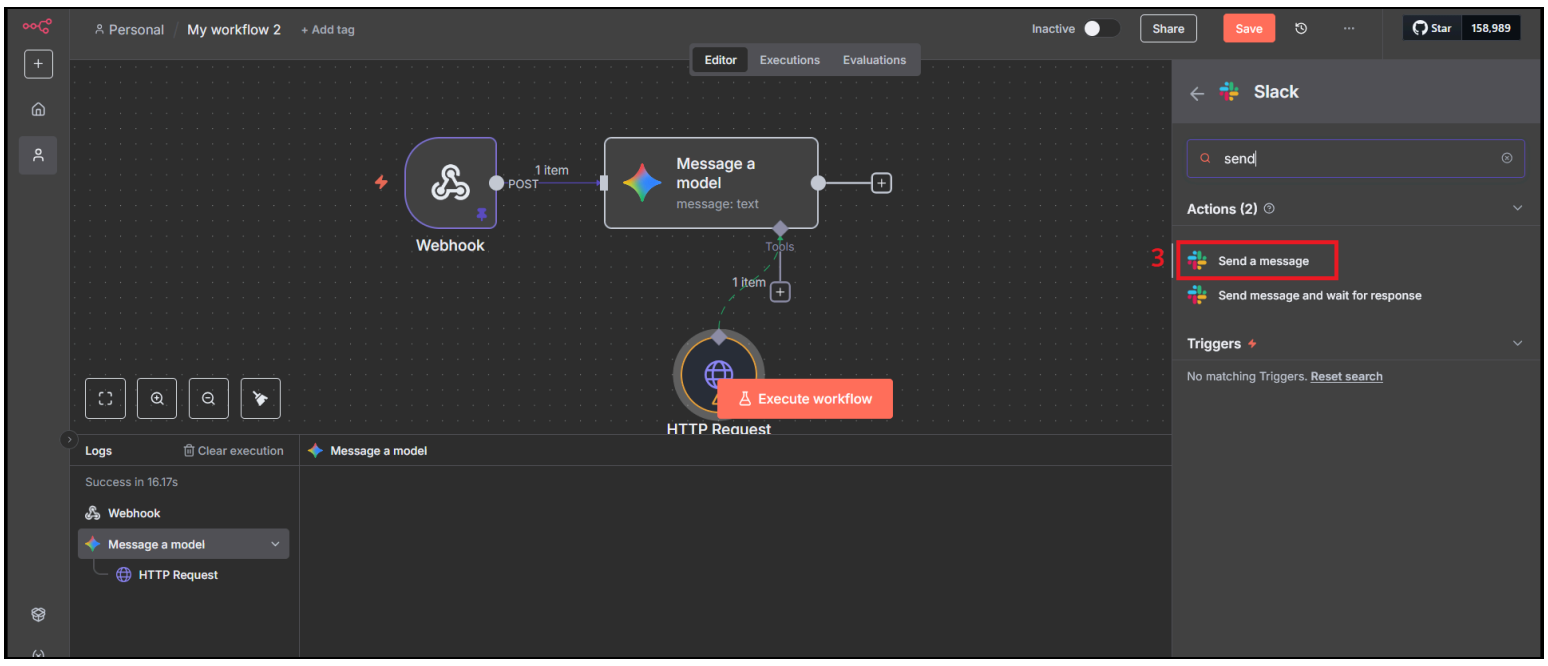
3. Name the channel, make it public, and click **Create**.



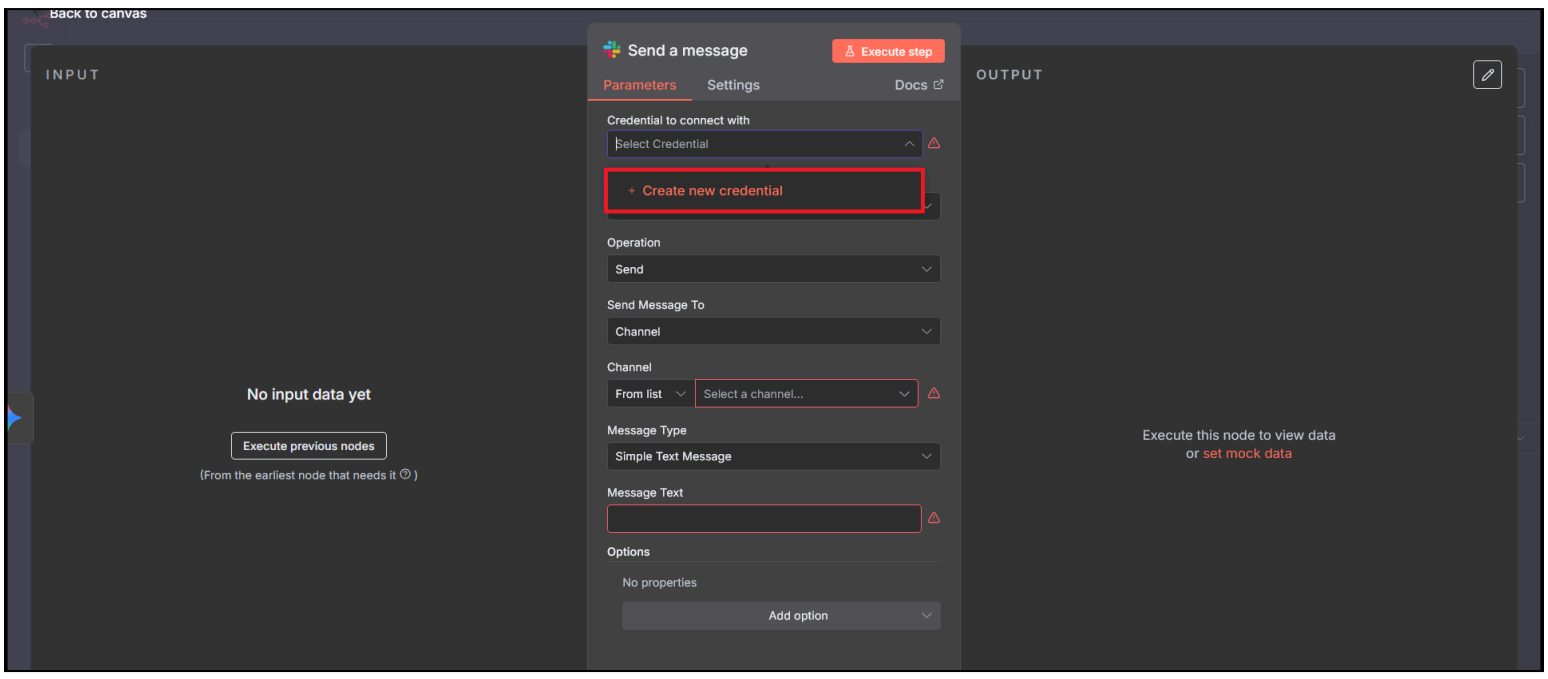
With the channel created (for example: **`alert-winlogon-fail`**), return to n8n:

1. Click the **+** icon next to **Message a Model**.
2. Search **Slack** and select **Send a Message**.

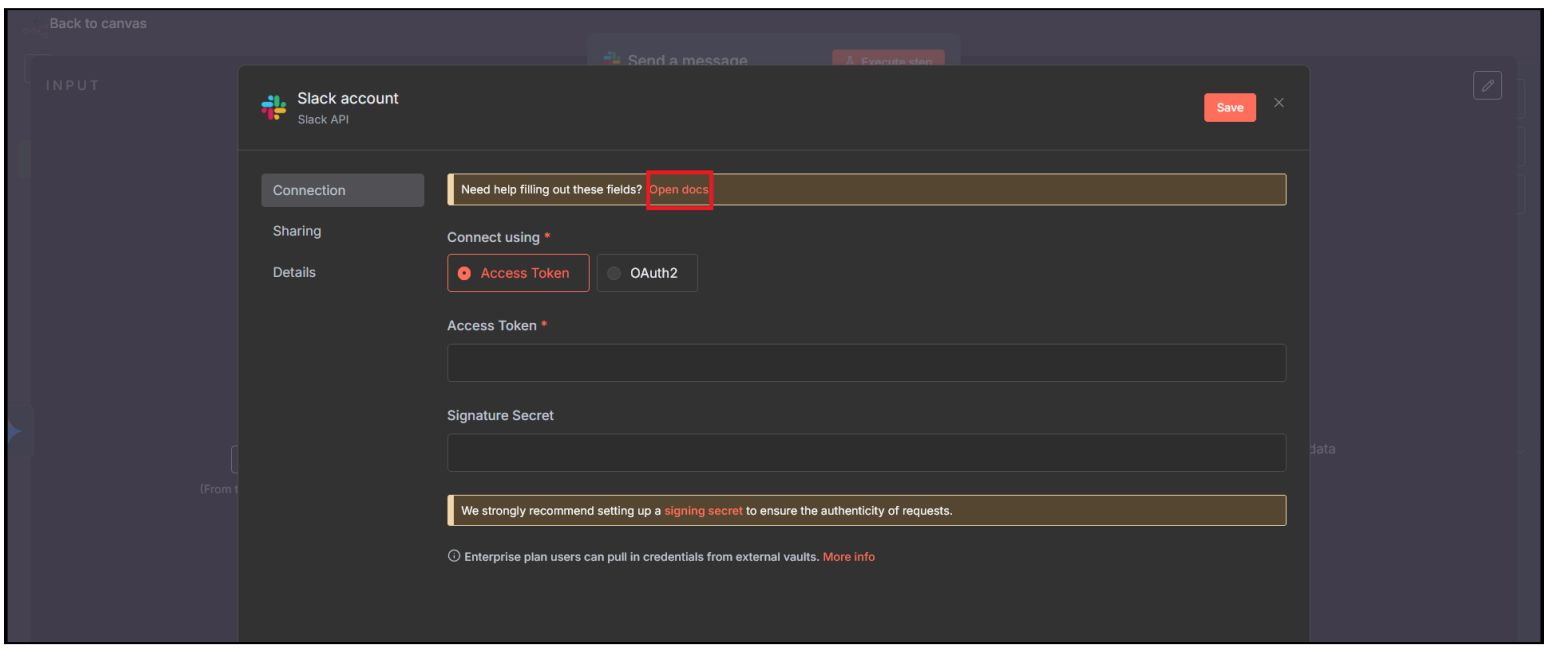


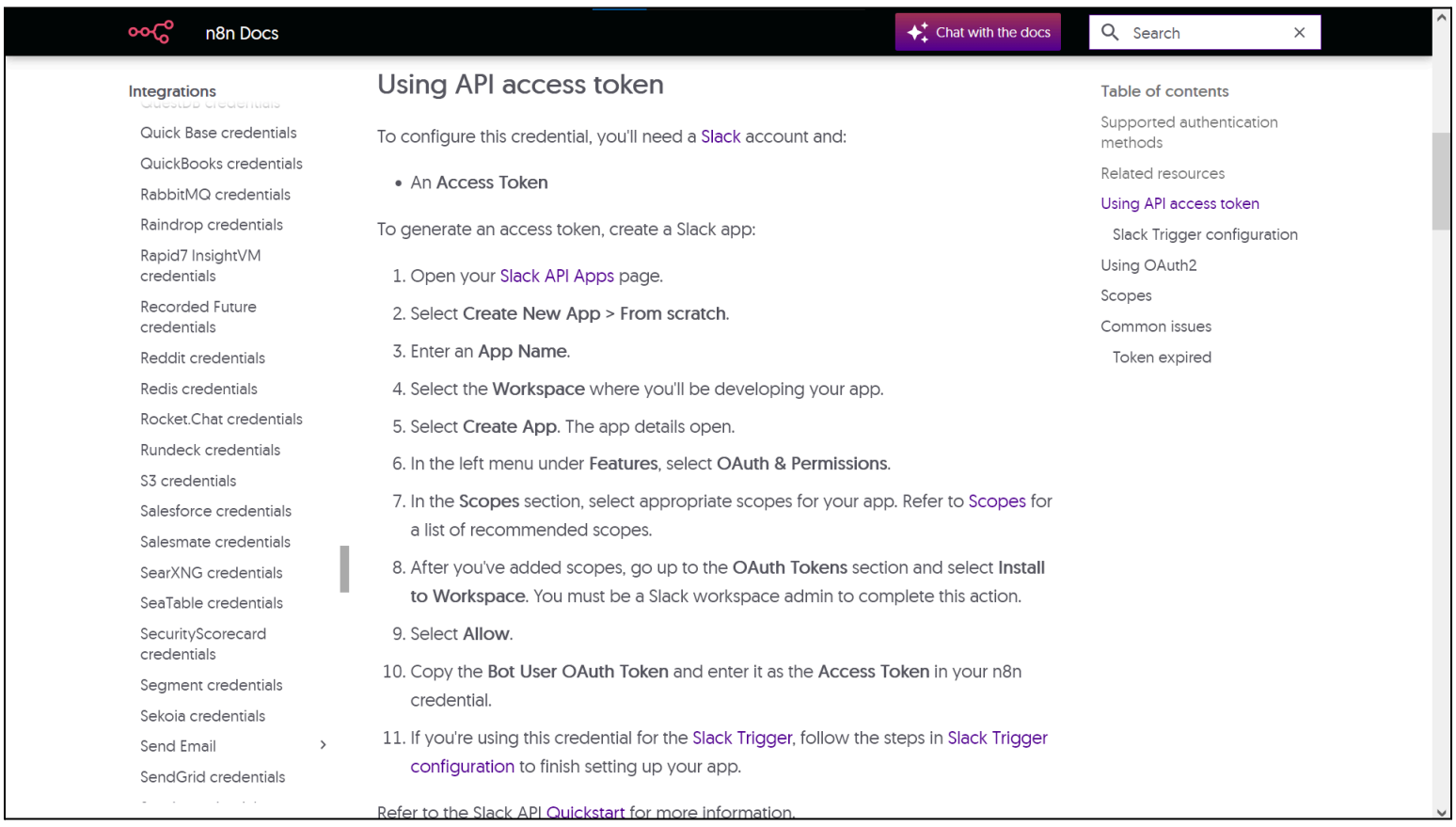


3. Under **Credentials**, choose **+ Create new credential**.



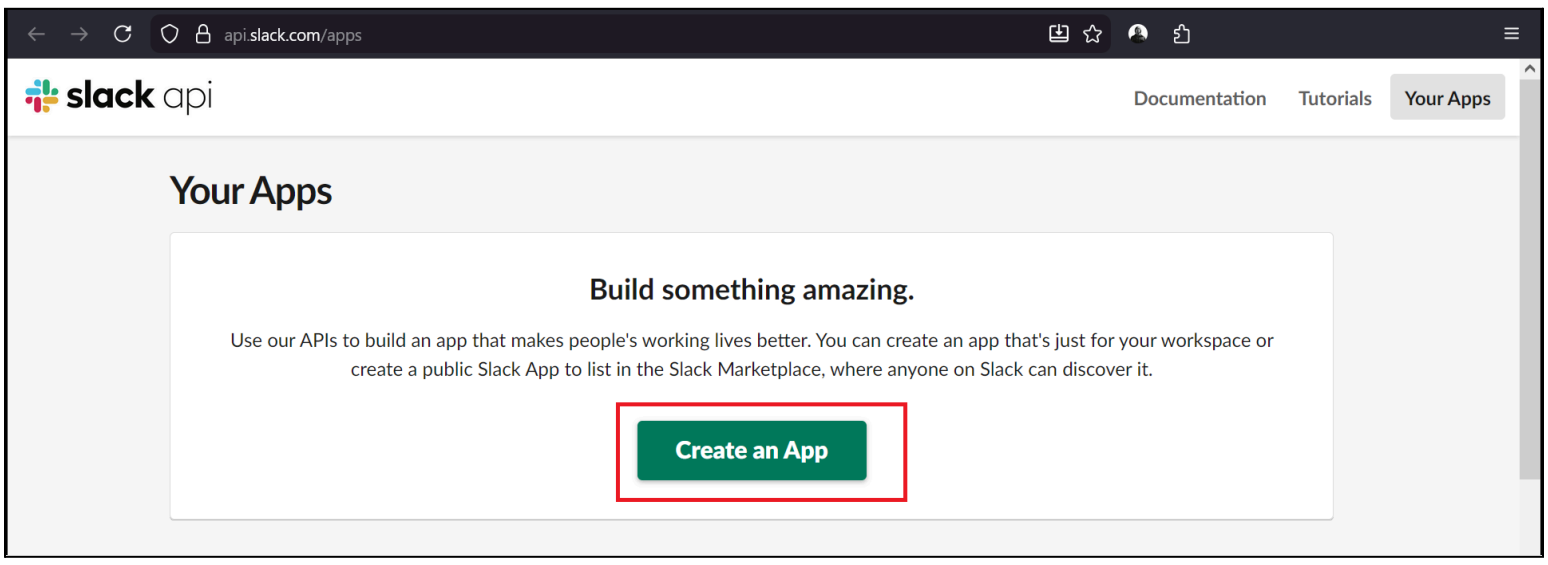
4. Click **Open Docs** and follow the steps:



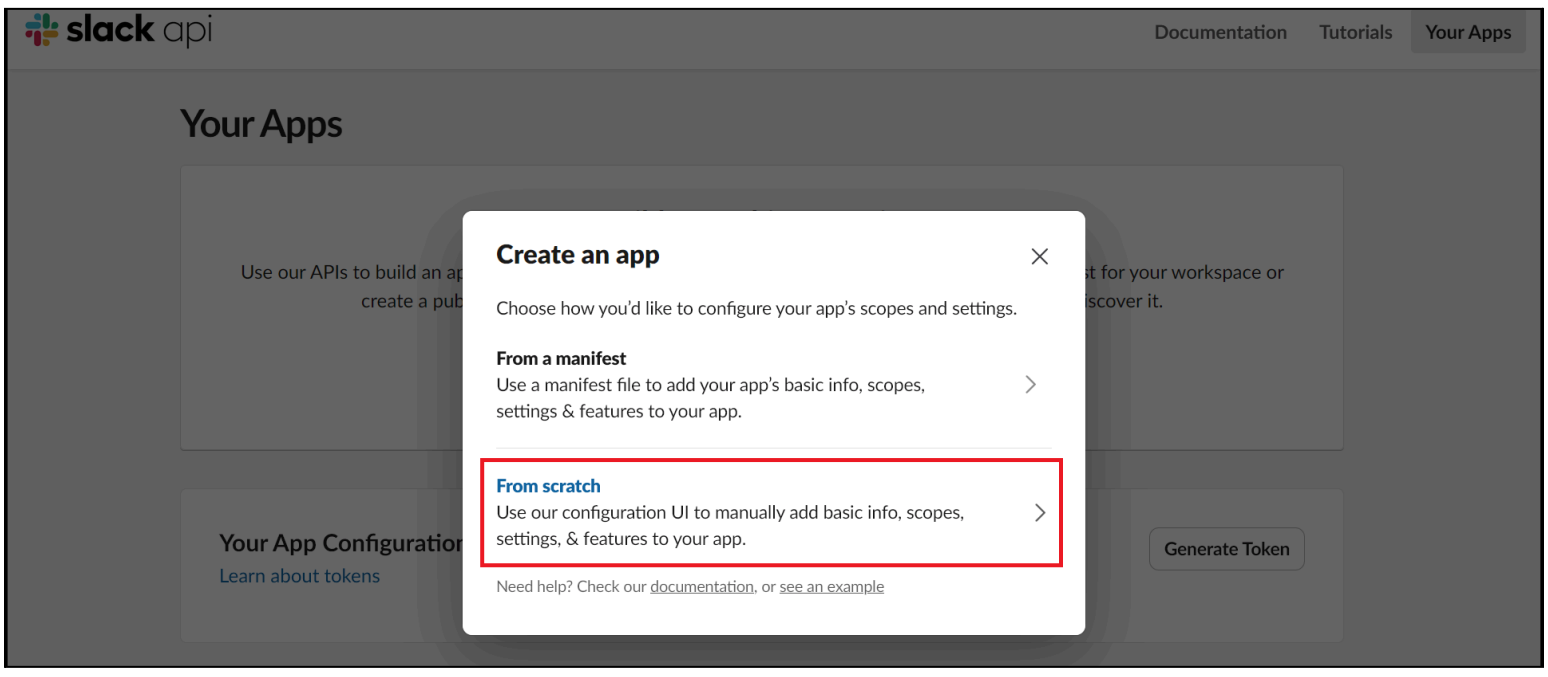


To create a Slack App:

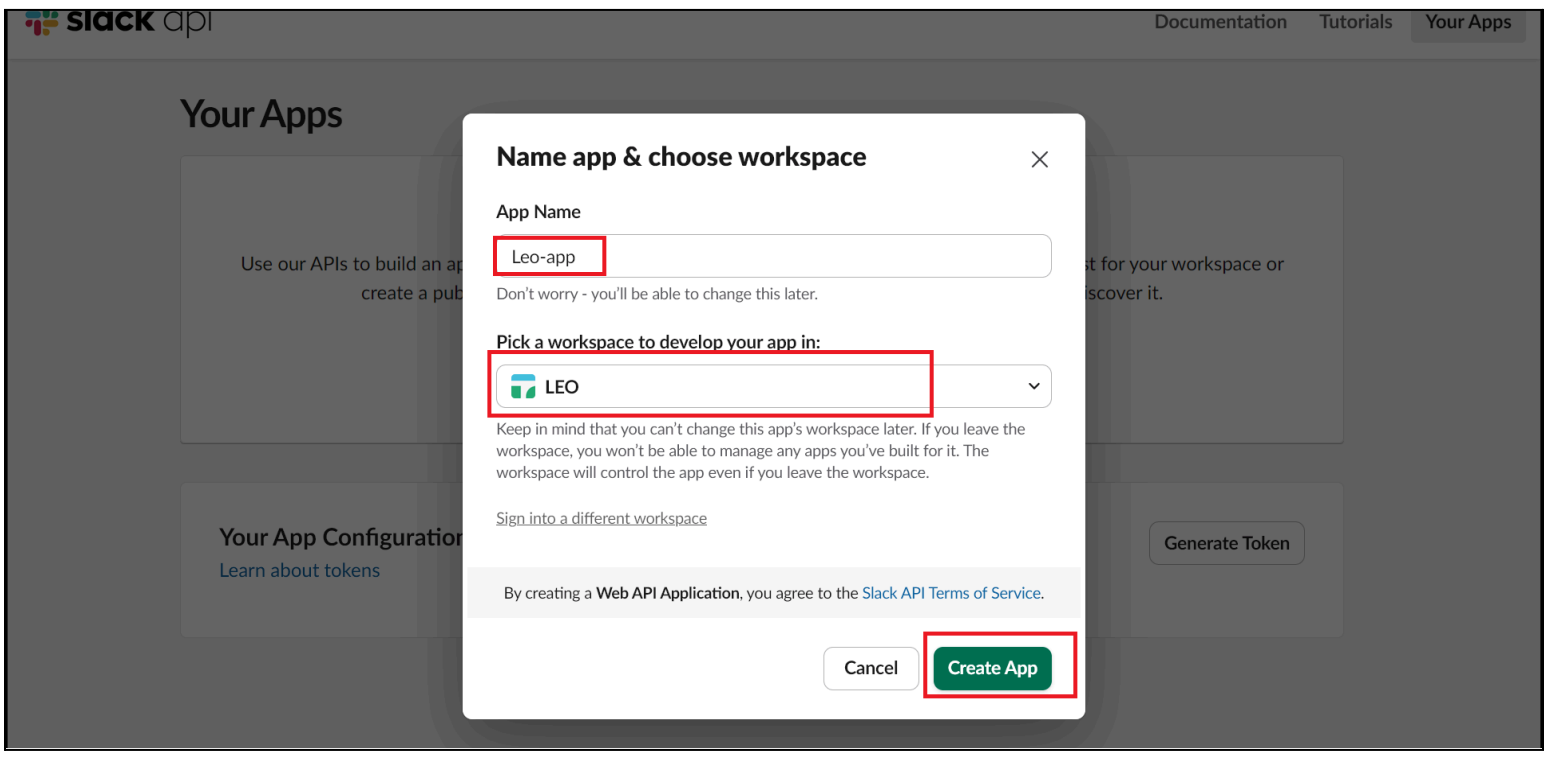
1. Click **Create an App**.



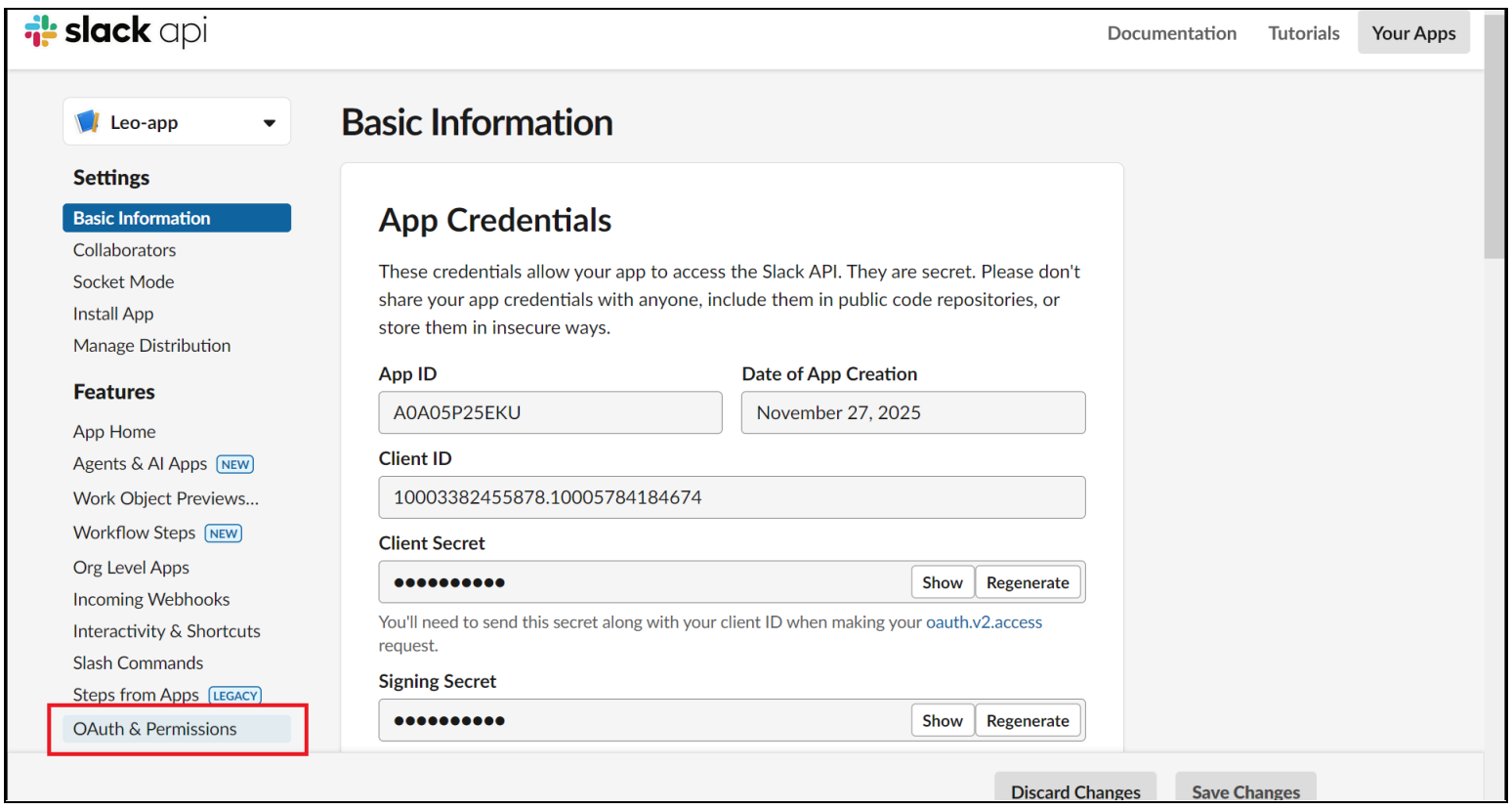
2. Choose **From Scratch**.



3. Name your app, select the same workspace where you created the channel, and click "**Create App**."



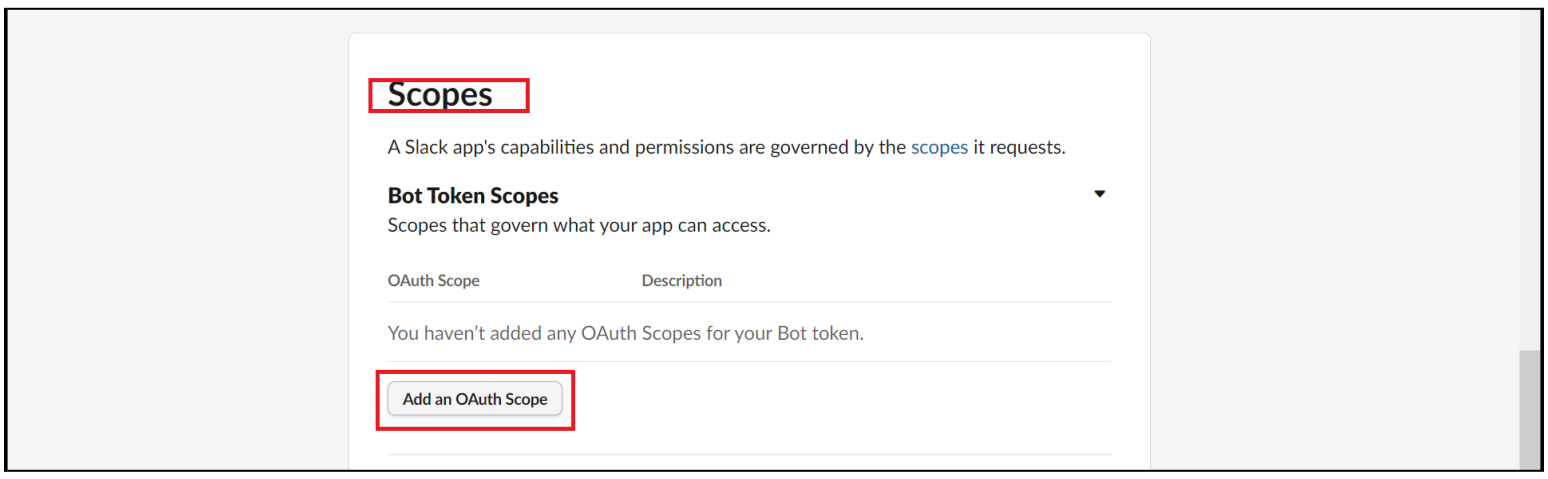
4. Open **OAuth & Permissions** from the left menu.



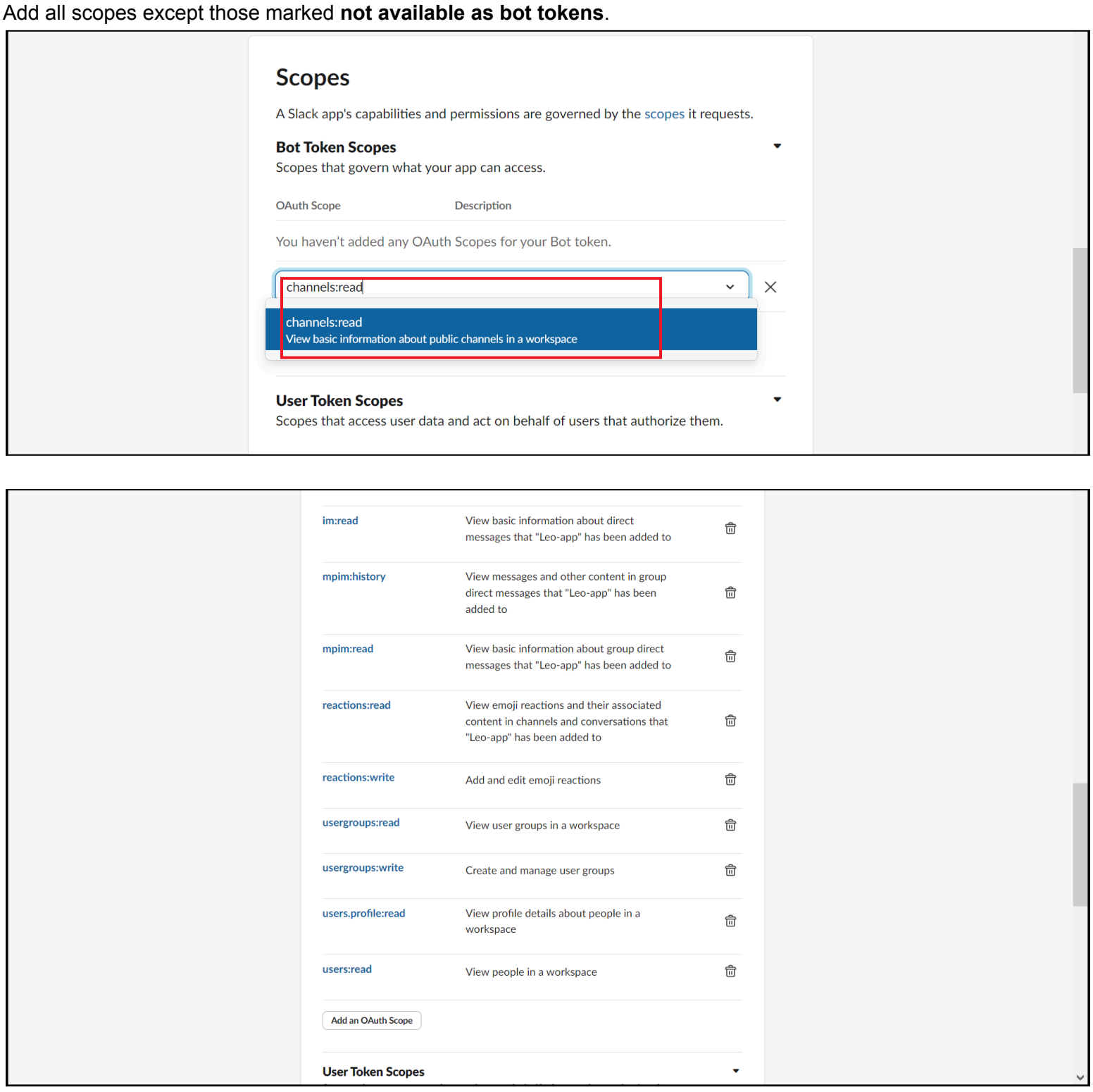
Add the recommended OAuth scopes:

1. In the documentation, copy each scope listed.

2. In Slack, click **Add an OAuth Scope** and paste each one.

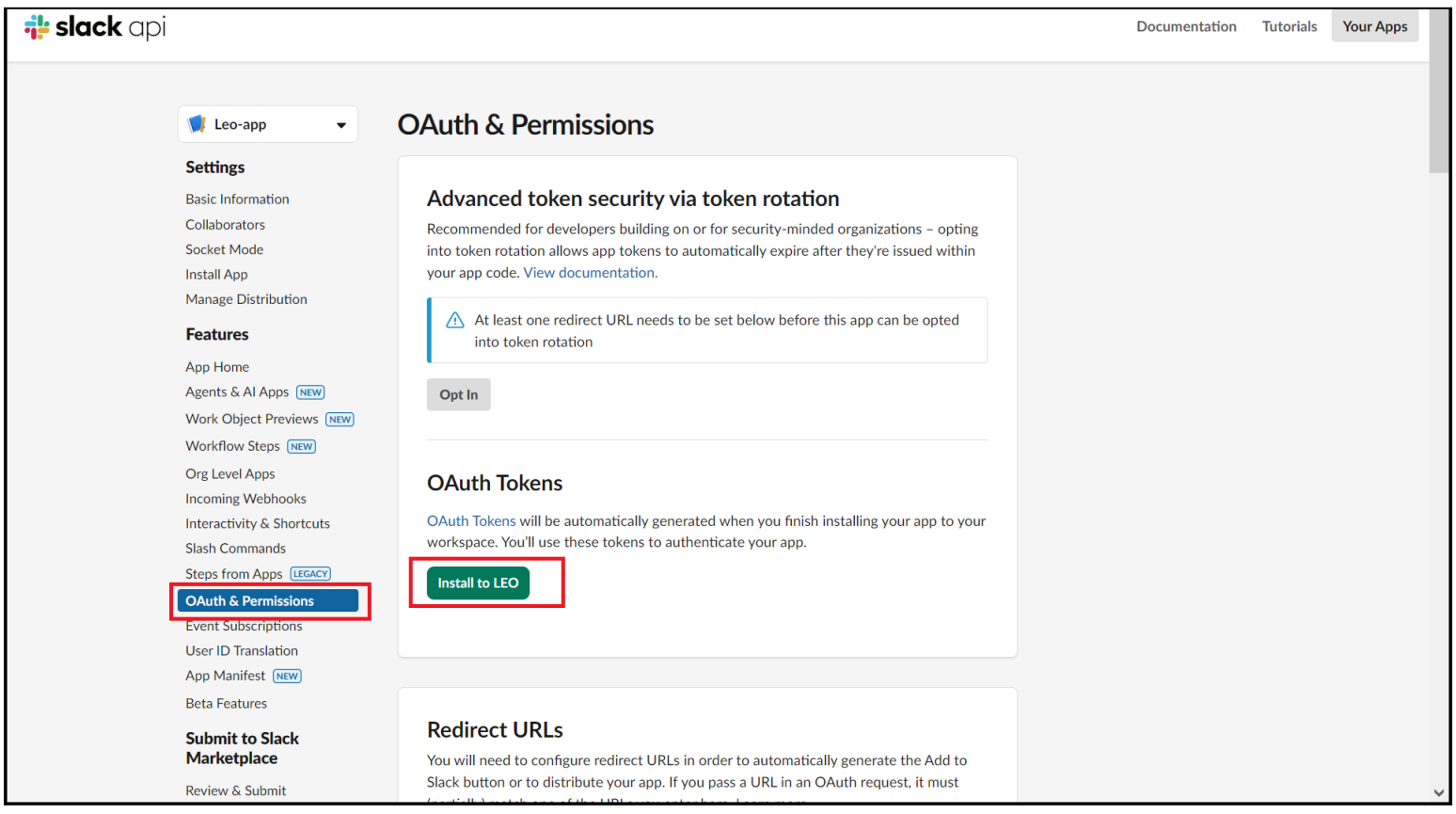


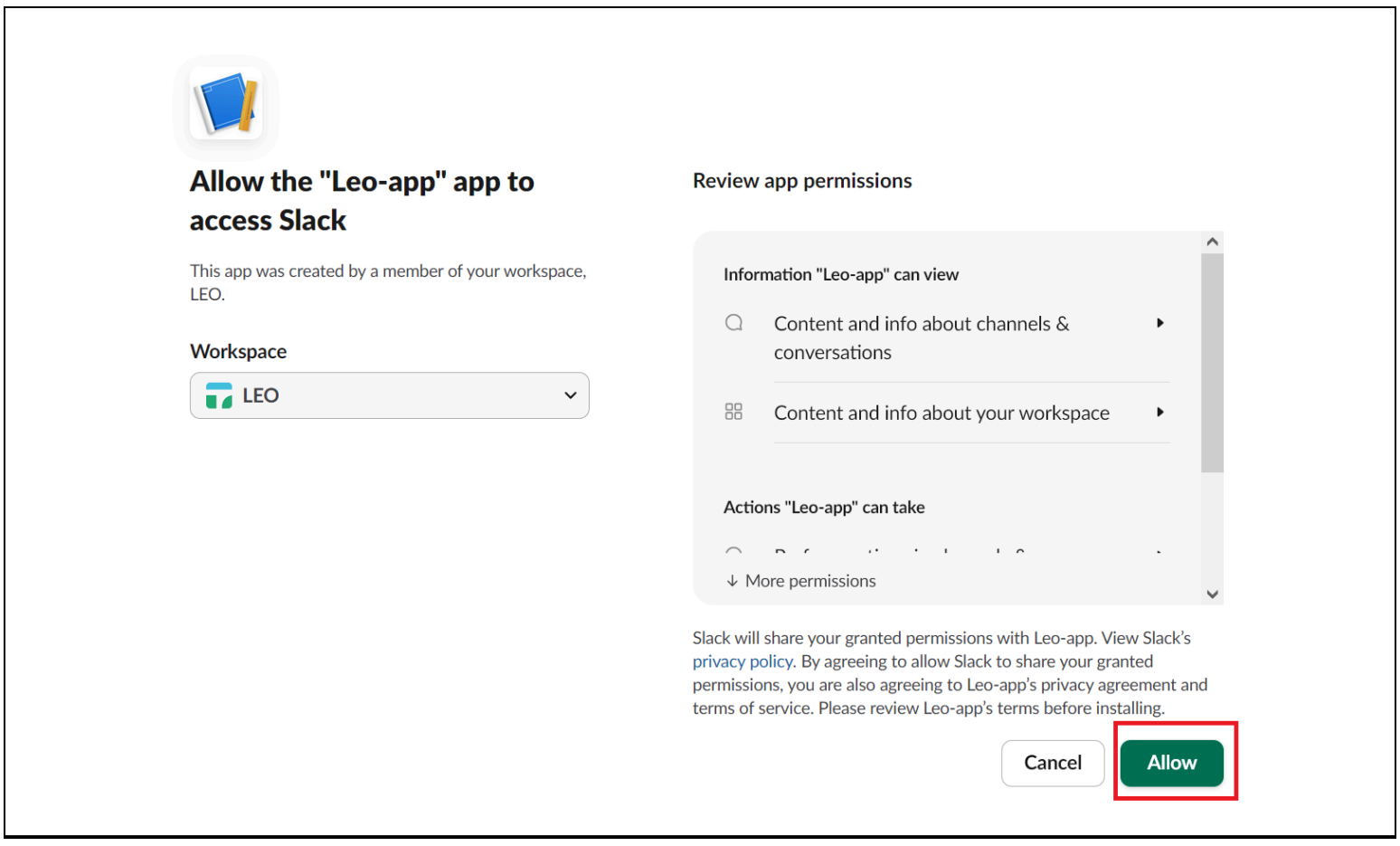
3. Add all scopes except those marked **not available as bot tokens**.



After adding all scopes:

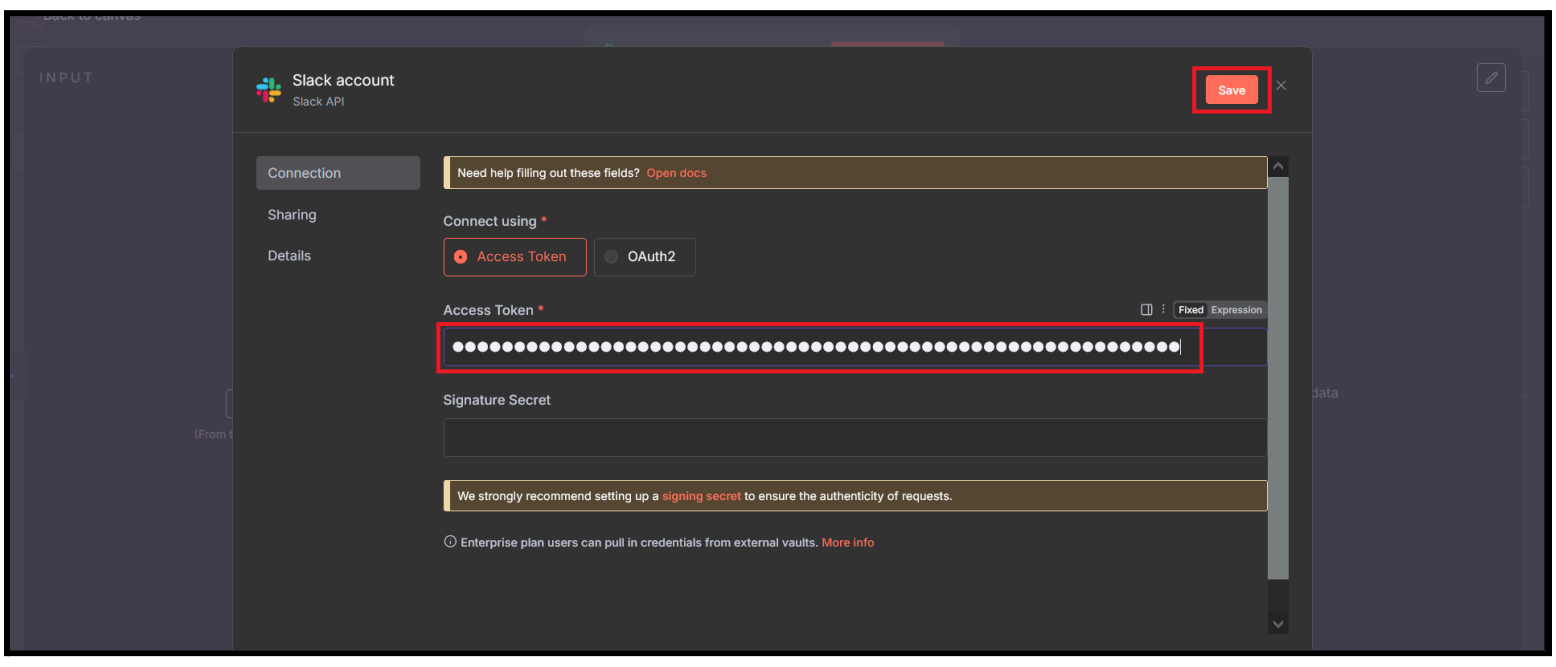
1. In the **OAuth & Permissions** page, scroll to **OAuth Tokens**.
2. Click **Install App to Workspace** and allow access.





3. Copy the **Bot User OAuth Token**.

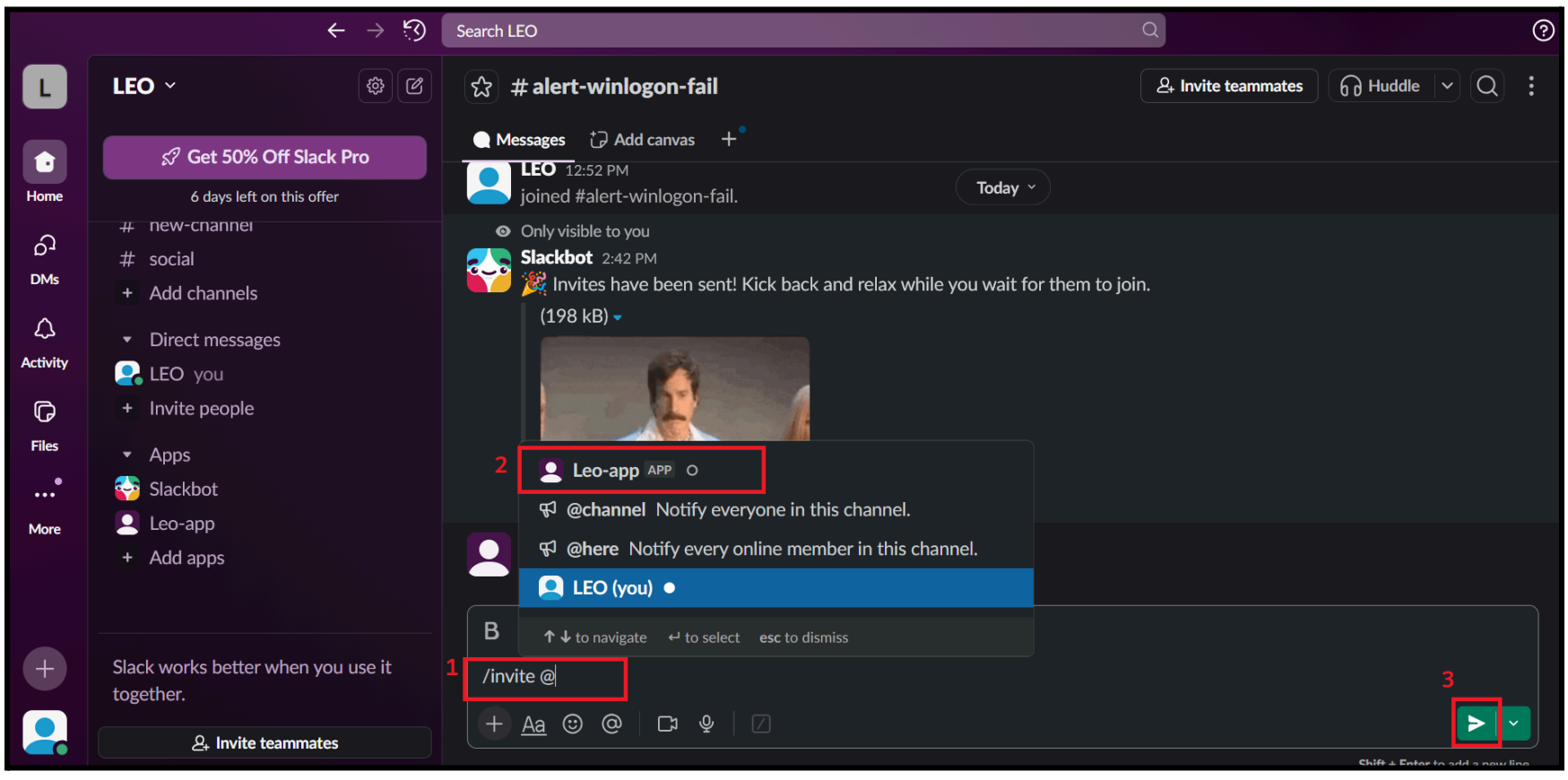
4. Paste it into the **Access Token** field in n8n and **save** the Slack credential.



Invite your Slack app to the channel:

```
/invite @your-app-name
```

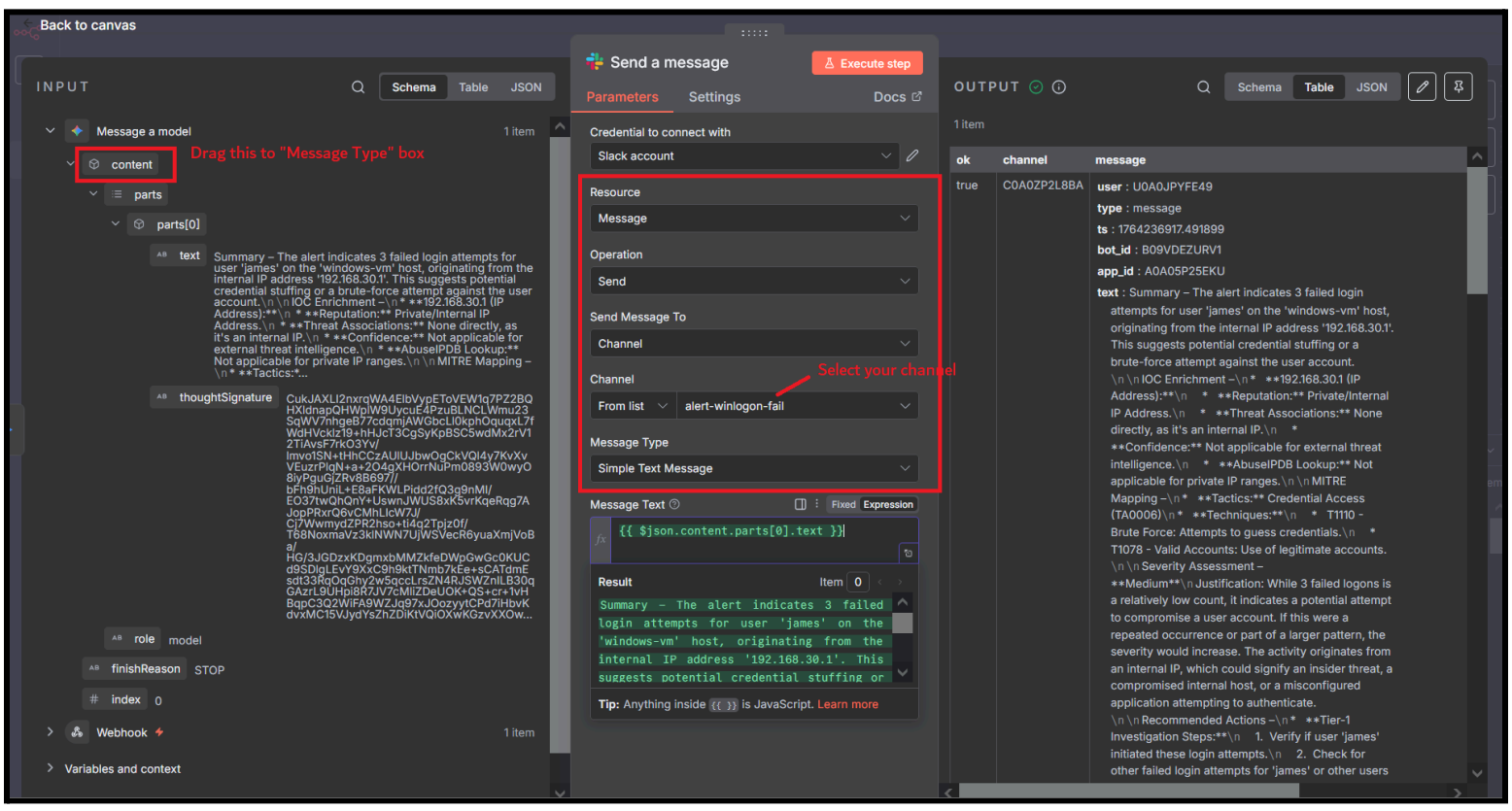
Example: **`/invite @Leo-app`**



Configure the Slack Node

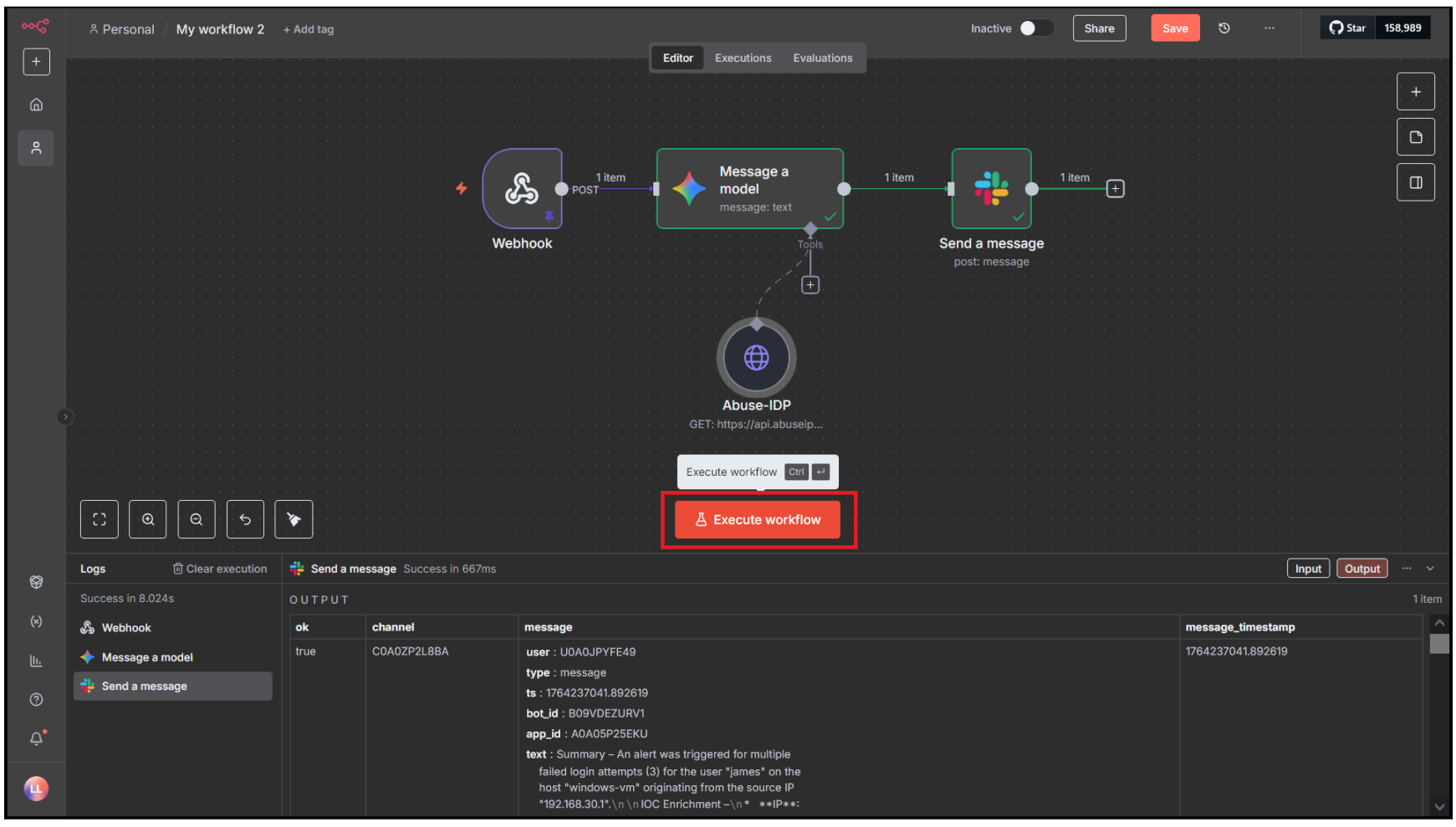
Back in n8n:

1. Set **Resource** to **Message**.
2. Set **Operation** to **Send**.
3. Under **Send Message To**, select **Channel**.
4. Choose your Slack channel (e.g., **`alert-winlogon-fail`**).
5. Drag the **content** output from your model into the **Message Text** field.

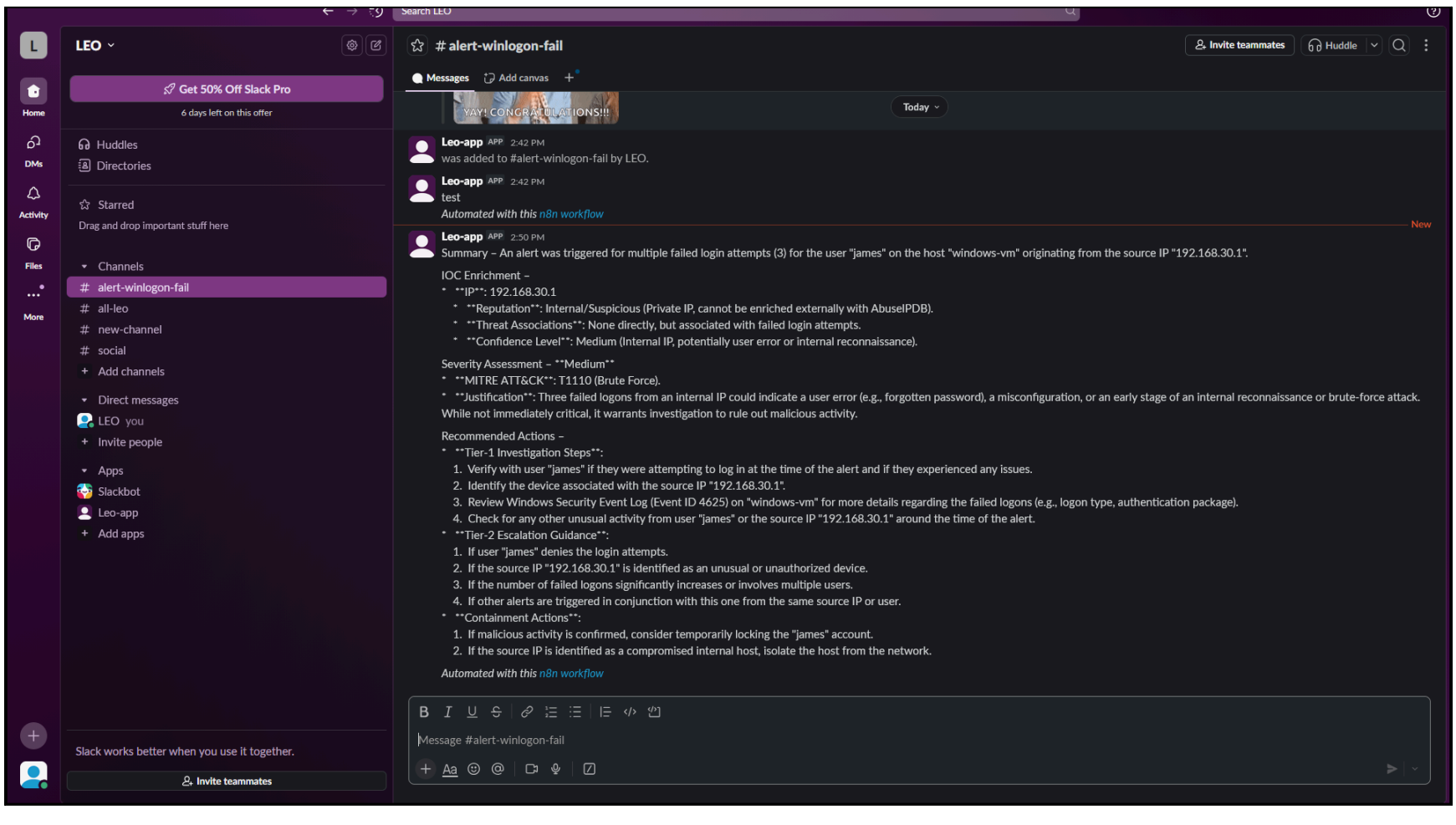


Execute the Workflow

Click **Execute Workflow**.



You should now see a Slack message containing the Gemini alert summary and recommended response actions.



At this point, your workflow is fully configured. Feel free to extend it with additional enrichment sources, custom logic, or downstream automations.