Ubuntu Pro for WSL

Canonical Ltd.

Jul 22, 2024

CONTENTS

> **Note**
>
> This documentation describes a future release of UP4W. UP4W is not yet generally available in the Microsoft Store.

Ubuntu Pro for WSL (UP4W) is a powerful automation tool for managing WSL instances from a Windows host. If you are responsible for a fleet of devices, UP4W will supercharge your ability to monitor, customise and secure WSL instances.

UP4W is designed to achieve close integration with applications for customising images, enforcing standards and validating security compliance. WSL instances can be created, removed and monitored with Landscape. Microsoft Defender is WSL-aware, making it easy to confirm if instances are compliant. Cloud-init support is built-in, allowing efficient customisation of standard images.

UP4W can be installed from the Microsoft Store and its setup is highly automated. Once you have an Ubuntu Pro subscription, adding your Pro token to UP4W on the Windows host will add that token to all connected WSL instances with the Ubuntu Pro client installed. When the Landscape client is installed on the host, any connected WSL instances will be auto-enrolled in Landscape. WSL instances can then be created, provisioned and managed from the Windows host.

WSL is preferred by many organisations as a solution to run a fully-functioning Linux environment on a Windows machine. UP4W empowers system administrators and corporate security teams to manage these WSL instances at scale.

CHAPTER

ONE

IN THIS DOCUMENTATION

Tutorial **Start here** with a hands-on introduction for new users, guiding you through your first-steps

How-to guides **Follow step-by-step** instructions for key operations and common tasks

Reference **Read technical descriptions** of important factual information relating to UP4W

UP4W Dev **Review guides and reference material** aimed at contributors

CHAPTER

TWO

PROJECT AND COMMUNITY

UP4W is a member of the Ubuntu family. It's an open-source project that warmly welcomes community contributions, suggestions, fixes and constructive feedback. Check out our contribution page on GitHub in order to bring ideas, report bugs, participate in discussions and much more!

Thinking about using UP4W for your next project? Get in touch!

2.1 Get started with UP4W

Windows Subsystem for Linux (WSL) makes it possible to run Ubuntu — the number 1 open-source operating system — on a Windows machine. With Ubuntu Pro for WSL (UP4W) an Ubuntu Pro subscription empowers you to manage Ubuntu WSL instances at scale.

In this tutorial you will develop an understanding of how UP4W can be installed and deployed for managing multiple WSL instances.

2.1.1 What you will do

- Install UP4W from the Microsoft Store
- Test automatic configuration of WSL instances by UP4W
- Manage WSL instances from Landscape

2.1.2 What you need

- A Windows 10 or 11 machine with a minimum of 16GB RAM and 8-core processor to host WSL instances
- The latest version of Landscape Server set up and configured on a physical or virtual machine

Note

WSL enables using a Linux shell and Windows PowerShell side-by-side on the same machine. Throughout this tutorial, commands will be prefixed by a prompt that indicates the shell being used, for example:

- `PS C:\Users\me\tutorial>` is a PowerShell prompt where the current working directory is `C:\Users\me\tutorial`.
- `ubuntu@wsl:~/tutorial$` indicates a Linux shell prompt login as ubuntu where the current working directory is `/home/ubuntu/tutorial/`

Output logs are included in this tutorial when instructive but are typically omitted to save space.

2.1.3 Set things up

Install WSL and Ubuntu

WSL can be installed directly from the Microsoft Store.

If you already have WSL installed, with `~\.wslconfig` on your system, you are advised to backup the file and remove it before continuing the tutorial.

To check if the file exists run:

```
PS C:\Users\me\tutorial> Test-Path -Path "~\.wslconfig"
```

If this returns `True` then the file exists and can be removed with:

```
PS C:\Users\me\tutorial> Remove-Item ~\.wslconfig
```

Ubuntu 24.04 LTS is recommended for this tutorial and can be installed from the Microsoft Store:

- Install Ubuntu 24.04 LTS from the Microsoft Store

Warning

If you already have Ubuntu WSL pre-installed:

We recommend that any Ubuntu WSL installed is exported then deleted. You can then install them as described in this tutorial. At the end of the tutorial you can re-import and restore your data.

Exporting, deleting and re-importing Ubuntu is achieved with the following commands:

```
PS C:\Users\me\tutorial> wsl --export Ubuntu-24.04 .\backup\Ubuntu-{version}.tar.gz

PS C:\Users\me\tutorial> wsl --unregister Ubuntu-24.04

PS C:\Users\me\tutorial> wsl --import Ubuntu-24.04 .\backup\Ubuntu-24.04 .\backup\
↪Ubuntu-24.04.tar.gz
```

You can now launch WSL instances of Ubuntu on your Windows machine.

For the remainder of the tutorial you will need to have a Landscape server set up and you should be able access your Landscape dashboard in a browser. Please refer to the Landscape documentation for setup and configuration instructions.

Get an Ubuntu Pro token

An active Ubuntu Pro subscription provides you with a token that can be added to the Ubuntu Pro client on WSL instances.

Your subscription token can be retrieved from the Ubuntu Pro Dashboard.

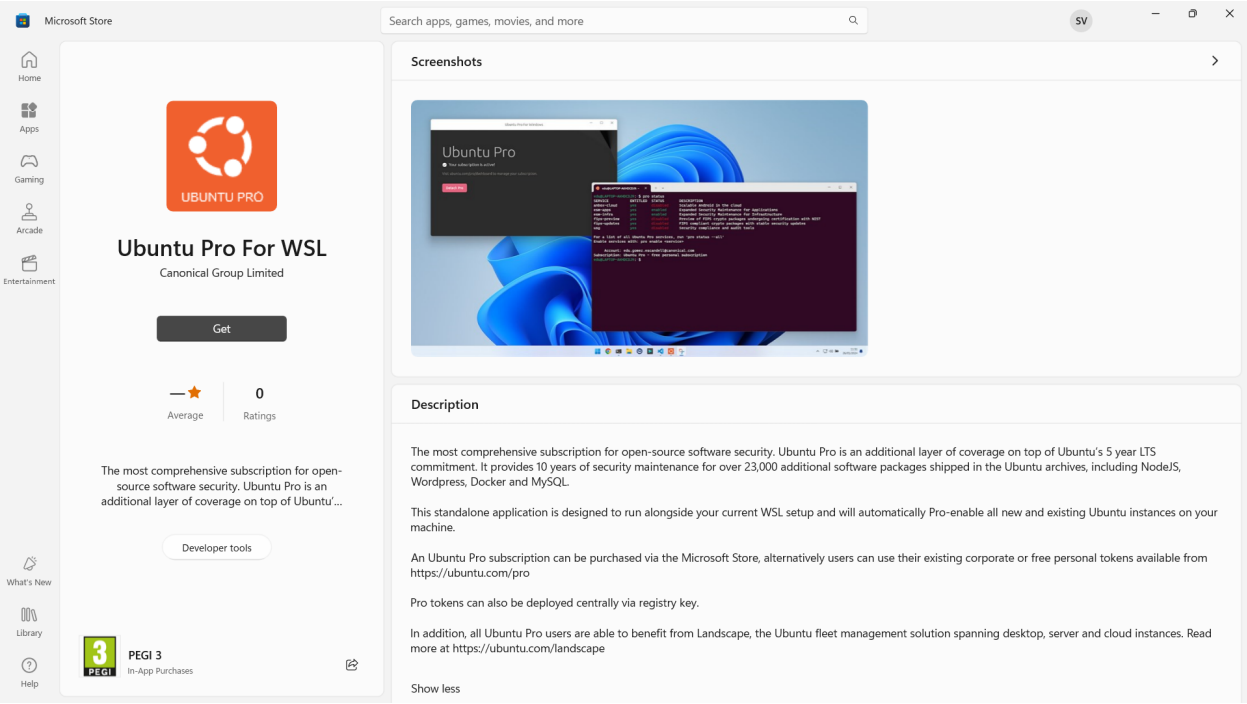
Visit the Ubuntu Pro page if you need a new subscription. The `Myself` option for a personal subscription is free for up to 5 machines.

Once you have a token you are ready to install UP4W.

Install UP4W

> **Warning**
>
> The install link below will work only if you're logged in to the Microsoft Store with an account for which access to the app has been enabled.

To install UP4W go to this link to the Microsoft Store and click **Install**.

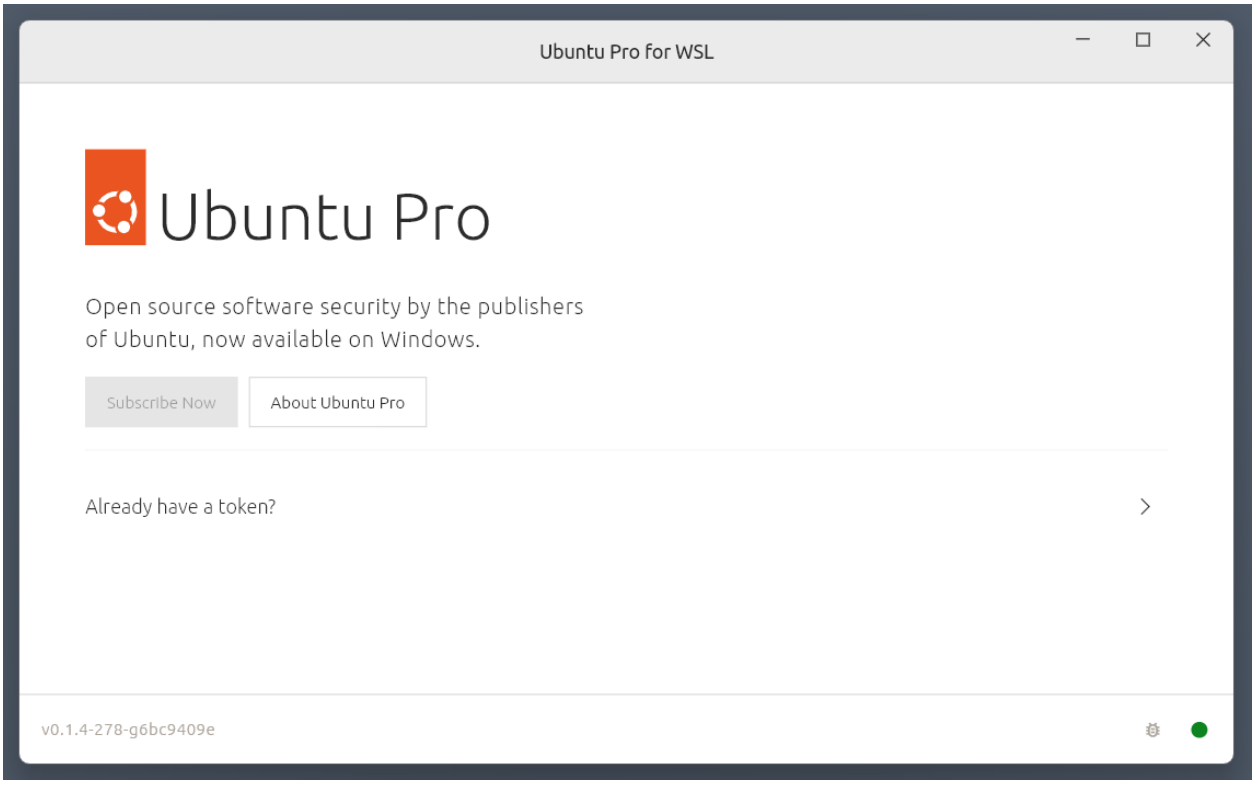


After installation has finished click **Start** to begin configuring UP4W.

> **Note**
>
> UP4W can be configured through the *Windows registry* instead of the GUI. The steps are outlined briefly in *Install and configure UP4W > Using the registry*. This approach may be more suitable when operating at scale.

In the UP4W Windows application click the arrow beside "Already have a token?".

Paste your token from the Ubuntu Pro dashboard during *Setup* and click **Confirm**. You will then be shown the Landscape configuration screen.



Create a new file in your home directory named `landscape.conf` and enter following contents. For the purpose of this tutorial, replace:

- <SERVER_URL> by the host url (or FQDN) of the machine that runs the Landscape server
- <YOUR_WINDOWS_USER_NAME> by the user name on the same Windows machine

```
[host]
url = <SERVER_URL>:6554
[client]
account_name = standalone
registration_key =
url = https://<SERVER_URL>/message-system
log_level = debug
ping_url = https://<SERVER_URL>/ping
```

Note

If the machine running the server is not trusted on your network you may need to explicitly reference a path to the SSL public key on a Windows host machine.

For example, if you followed the Landscape Quickstart installation, the auto-generated self-signed certificate can be found at `/etc/ssl/certs/landscape_server.pem`.

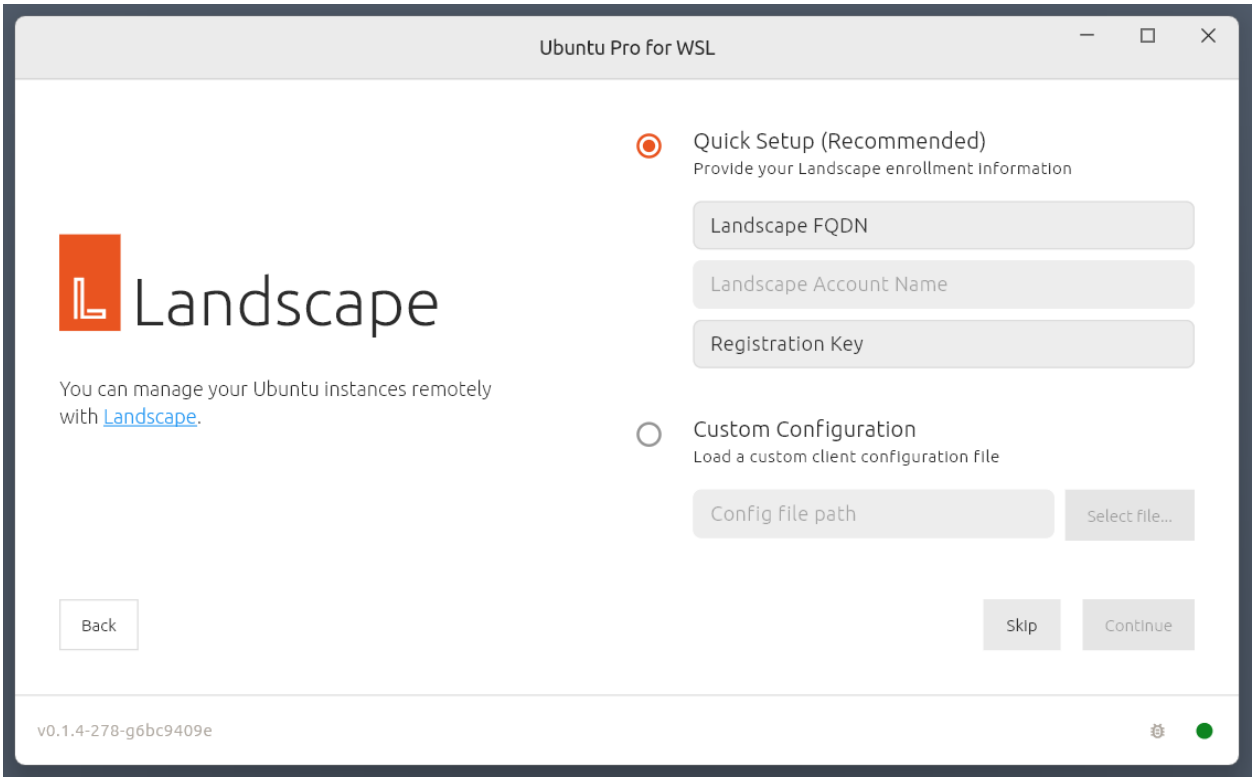
This can be copied to a Windows machine and referenced in `landscape.conf`:

ssl_public_key = C:\Users<YOUR_WINDOWS_USER_NAME>\landscape_server.pem

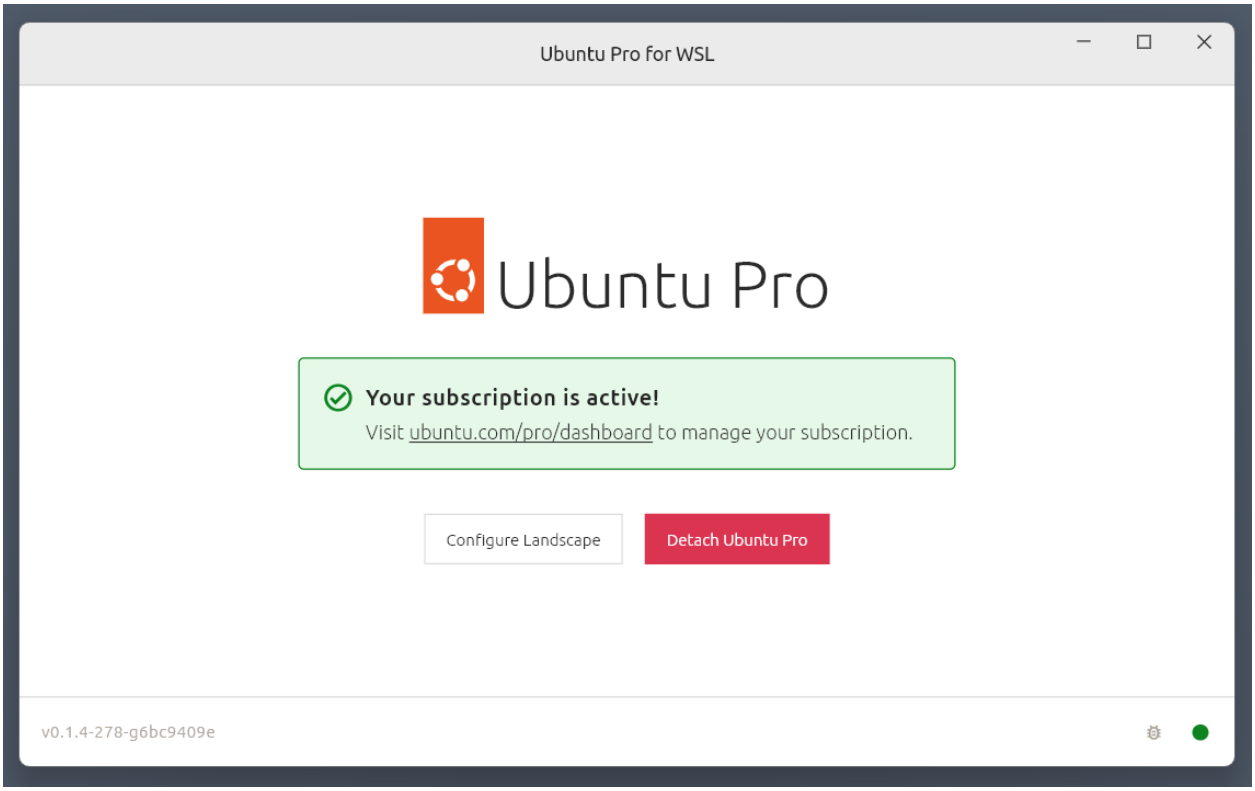
If necessary, the SSL public key can be added after the Windows host has first been registered in Landscape.

Then input <SERVER_URL> in “Quick Setup” in the field labelled “Landscape FQDN”. Alternatively, provide a path to

`landscape.conf` in “Custom Configuration”.



Click on the **Continue** button and you will see a status screen confirming the configuration is complete.



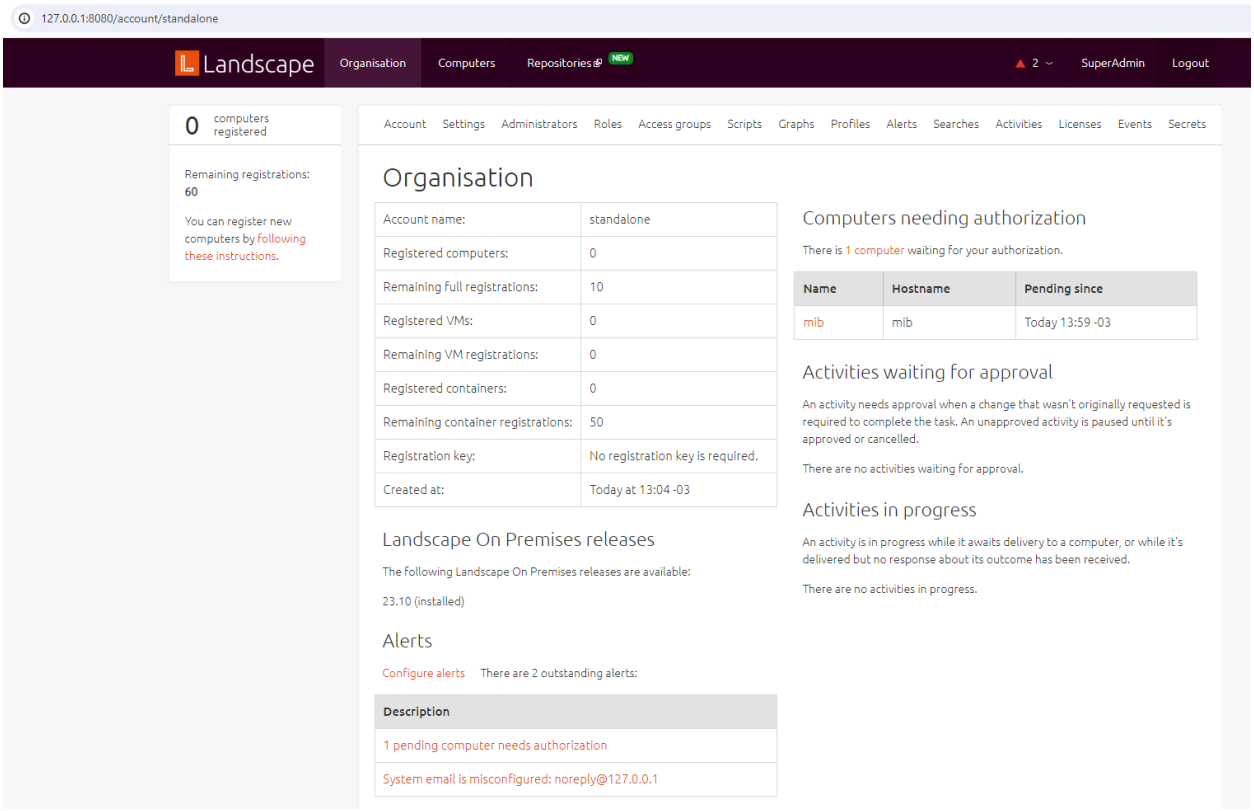
Done! You can now close the UP4W window.

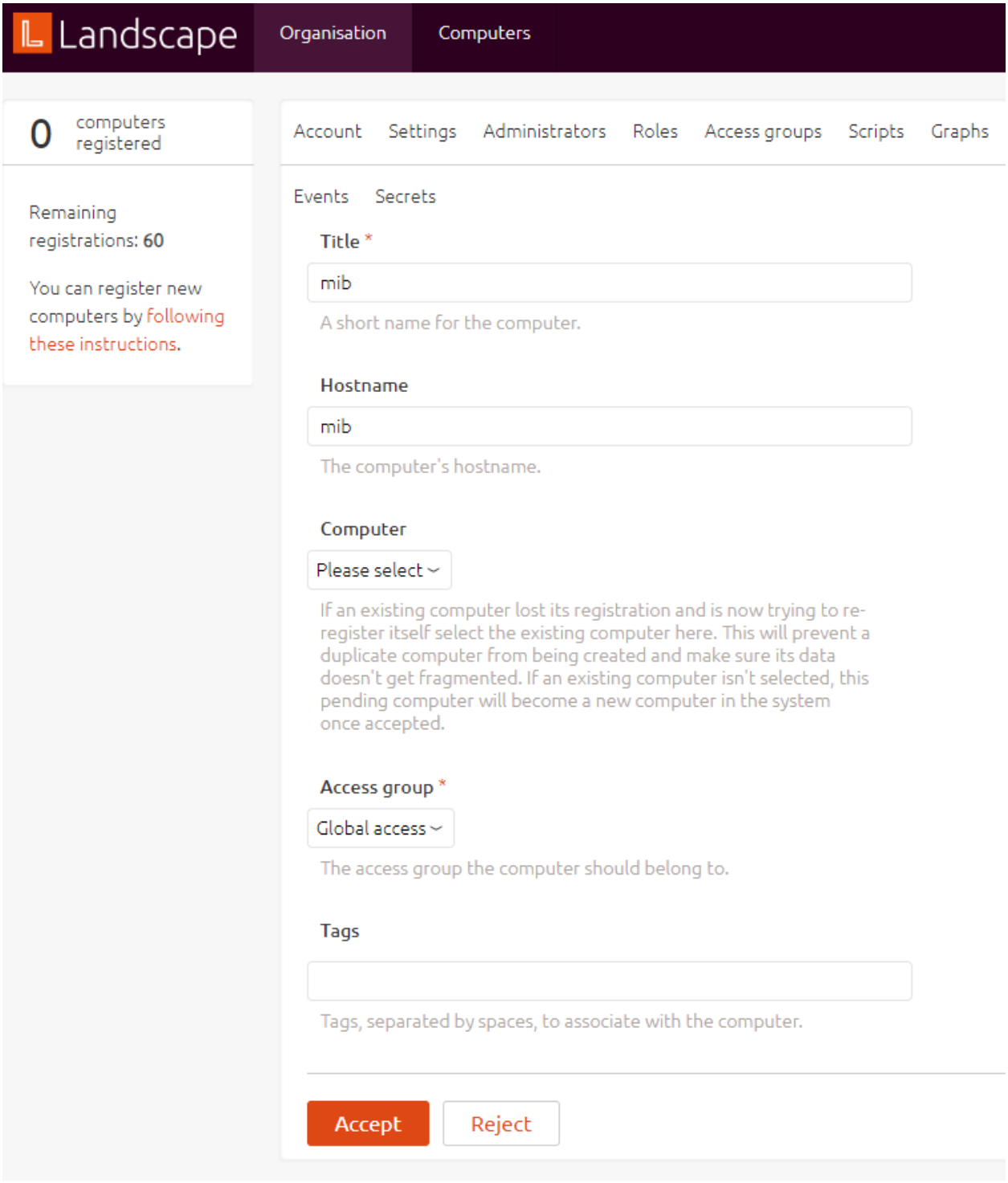
Your Ubuntu Pro subscription is now attached to UP4W on the Windows host. UP4W will automatically forward the subscription to the Ubuntu Pro client on your Ubuntu WSL instances. This means that all Ubuntu WSL instances will be automatically added to your Ubuntu Pro subscription.

This has also configured the Landscape client built into your UP4W Windows agent to know about your Landscape server; UP4W will forward this configuration to the Landscape client on your Ubuntu WSL instances as well; and all systems where the Landscape client has been configured this way are automatically registered with Landscape.

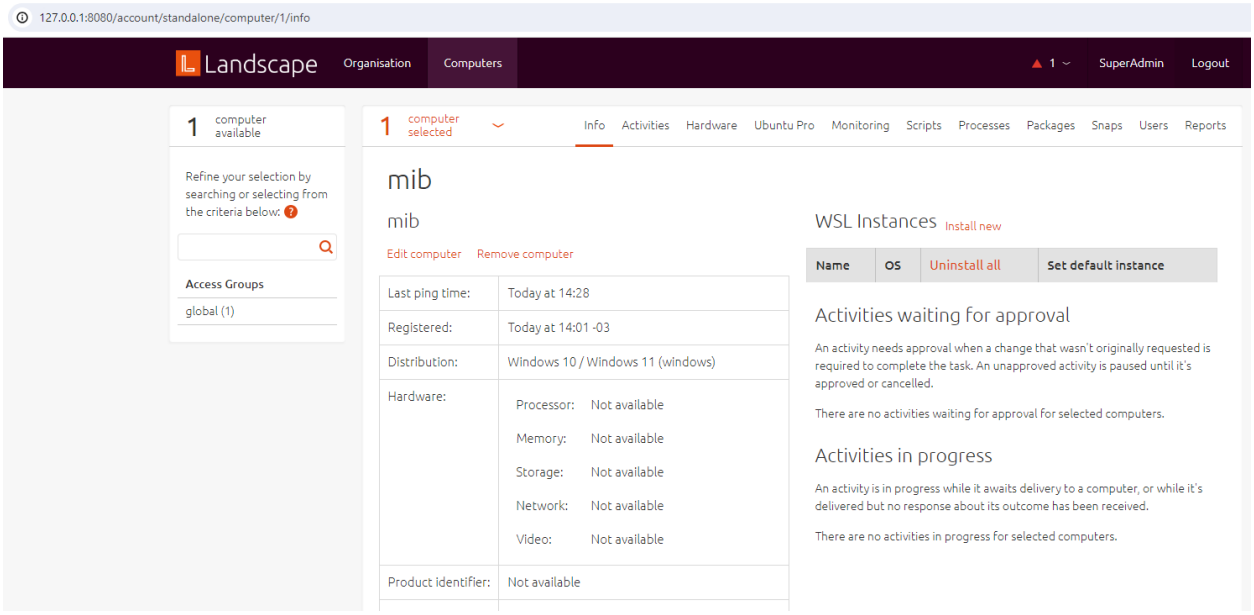
UP4W host registration with Landscape

Go back to your web browser and refresh the Landscape dashboard. On the right-hand side of the page you should see a request to approve your Windows host registration ("Computers needing authorisation"). Click on the computer name (in this case: `mib`) and when the new page loads click **Accept**.





At the top of the page, on the right-hand side of the Landscape logo, click on “Computers”. You should see your host machine listed. Details such as the operating system may take some time to appear.



Now you can leverage UP4W from your Landscape server to create and provision Ubuntu WSL instances on the host.

2.1.4 Deploy WSL instances

Create an Ubuntu WSL instance locally

Open Windows PowerShell and run the following command to create a new Ubuntu 24.04 instance. When prompted create the default user and password. For convenience, we'll set both to u.

```
PS C:\Users\me\tutorial> ubuntu2404.exe

Installing, this may take a few minutes...
Please create a default UNIX user account. The username does not need to match your␣
↪Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: u
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

u@mib:~$
```

You will now be logged in to the new instance shell and can check that UP4W has Pro-attached this instance:

The output indicates that services like ESM are enabled, with account and subscription information also shown:

```
u@mib:~$ pro status
SERVICE          ENTITLED  STATUS       DESCRIPTION
esm-apps         yes       enabled      Expanded Security Maintenance for Applications
esm-infra        yes       enabled      Expanded Security Maintenance for Infrastructure
```

(continues on next page)

(continued from previous page)

```
NOTICES
Operation in progress: pro attach

For a list of all Ubuntu Pro services, run 'pro status --all'
Enable services with: pro enable <service>

     Account: me@ubuntu.com
Subscription: Ubuntu Pro - free personal subscription
u@mib:~$
```

Packages can be accessed from all the enabled services:

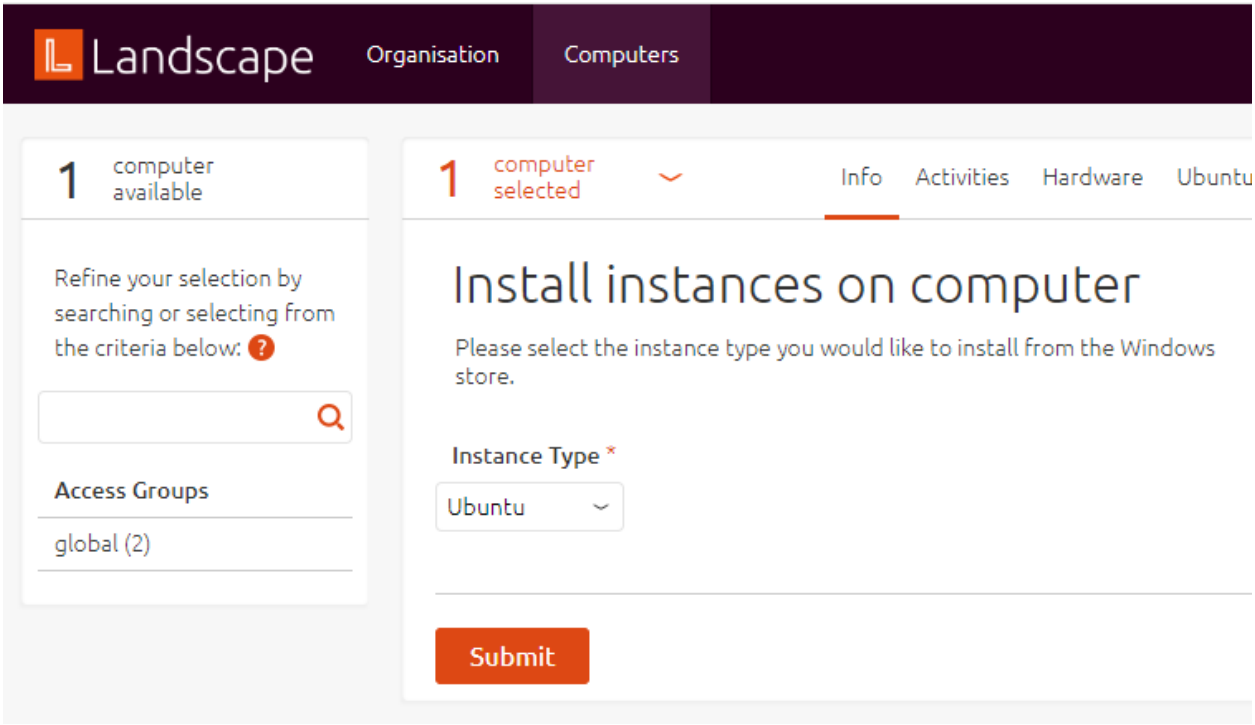
```
u@mib:~$ sudo apt update
Hit:1 http://archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ppa.launchpad.net/ubuntu-wsl-dev/ppa/ubuntu noble InRelease
Hit:3 http://security.ubuntu.com/ubuntu noble-security InRelease
Hit:4 http://archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:5 http://ppa.launchpad.net/landscape/self-hosted-beta/ubuntu noble InRelease
Hit:6 https://esm.ubuntu.com/apps/ubuntu noble-apps-security InRelease
Hit:7 http://archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:8 http://ppa.launchpad.net/cloud-init-dev/proposed/ubuntu noble InRelease
Hit:9 https://esm.ubuntu.com/infra/ubuntu noble-infra-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
```

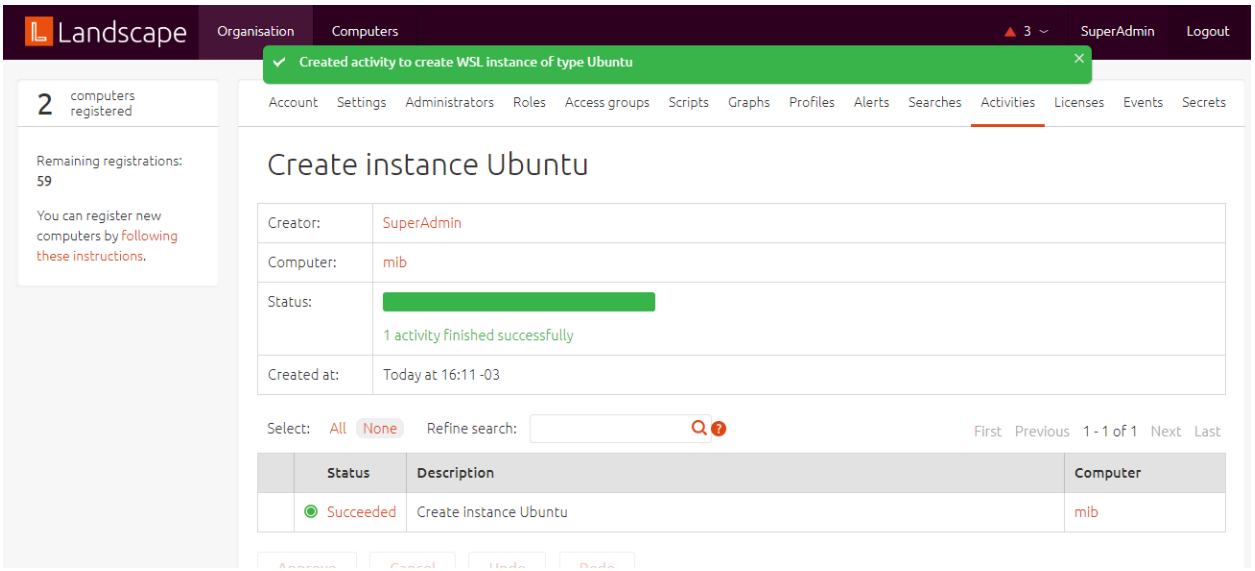
UP4W should have also Landscape-registered this instance. To verify, refresh the Landscape server web page and the instance should be listed under “Computers needing authorisation”.

To accept the registration click on the instance name, set “Tags” to `wsl-vision` in the pop-up then click **Accept**. The `wsl-vision` tag will be used for all the instances accepted into Landscape.

Create an Ubuntu WSL instance remotely

Back on the Landscape page in your web browser, navigate to “Computers” and click on the Windows machine (below: `mib`). You will find “WSL Instances” on the right side of the page. Click on the “Install new” link then set “Instance Type” to “Ubuntu” and click **Submit**. A status page will appear showing the progress of the new instance creation.





The Landscape server will talk to the Landscape client built into your UP4W. UP4W will then install the `Ubuntu` application and create an Ubuntu WSL instance automatically. In PowerShell, run `ubuntu.exe` to log in to the new instance.

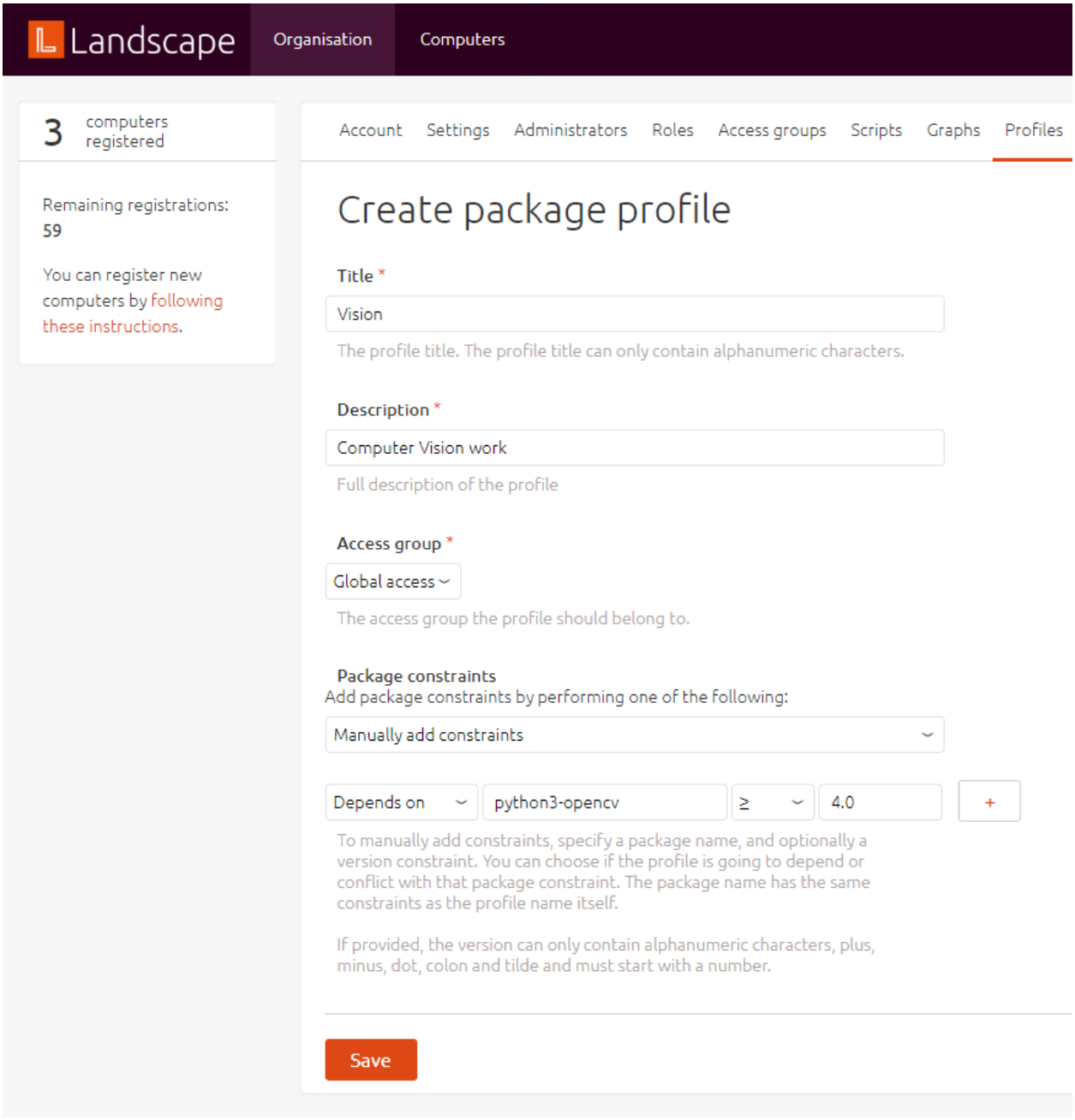
```
PS C:\Users\me\tutorial> ubuntu.exe

me@mib:~$
```

You can run `pro status` to verify pro-attachment and refresh your Landscape server page to verify and accept the registration. As before, apply the `wsl-vision` tag and click `Accept`.

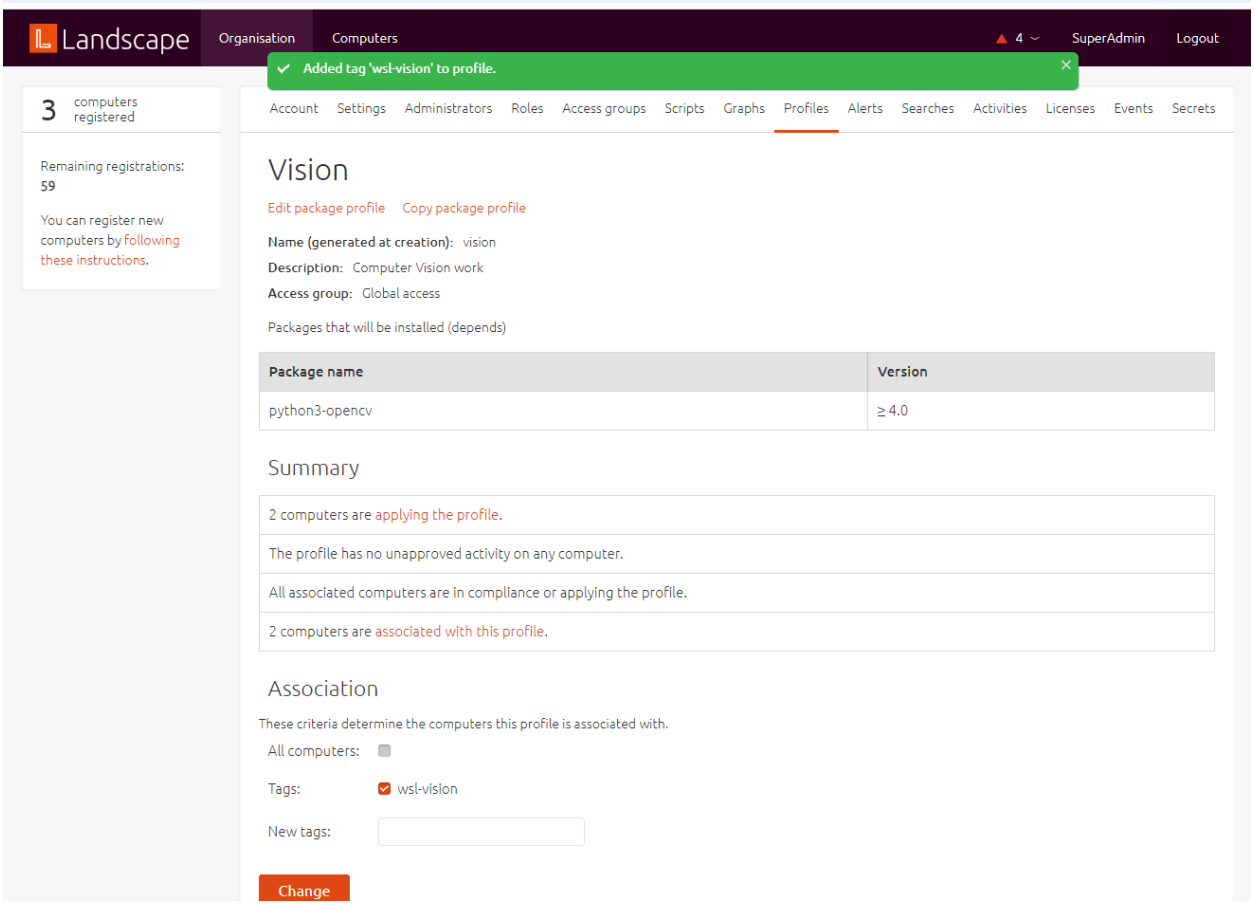
Deploy packages to all Ubuntu WSL instances

On your Landscape server page, navigate to `Organization > Profiles`, click on `Package Profiles` then `Add package profile`. Fill in the form with the following values and click “Save”.

Field	Value
Title	Vision
Description	Computer Vision work
Access group	Global
Package constraints	Manually add constraints
	Depends on `python3-opencv` >= `4.0`



On the bottom of the "Vision" profile page, in the "Association" section, set the "New tags" field to `wsl-vision` and click **Change**.



In the "Summary" section in the middle of the page you will see a status message showing that two computers are `applying the profile`. Click on the `applying the profile` link and then, in the "Activities" list, click on **Apply package profile** to see the progress of the package deployment.

When this process has completed, use one of your instance shells to verify that the `python3-opencv` package has been installed. For example, in the `Ubuntu` instance the first three packages returned are:

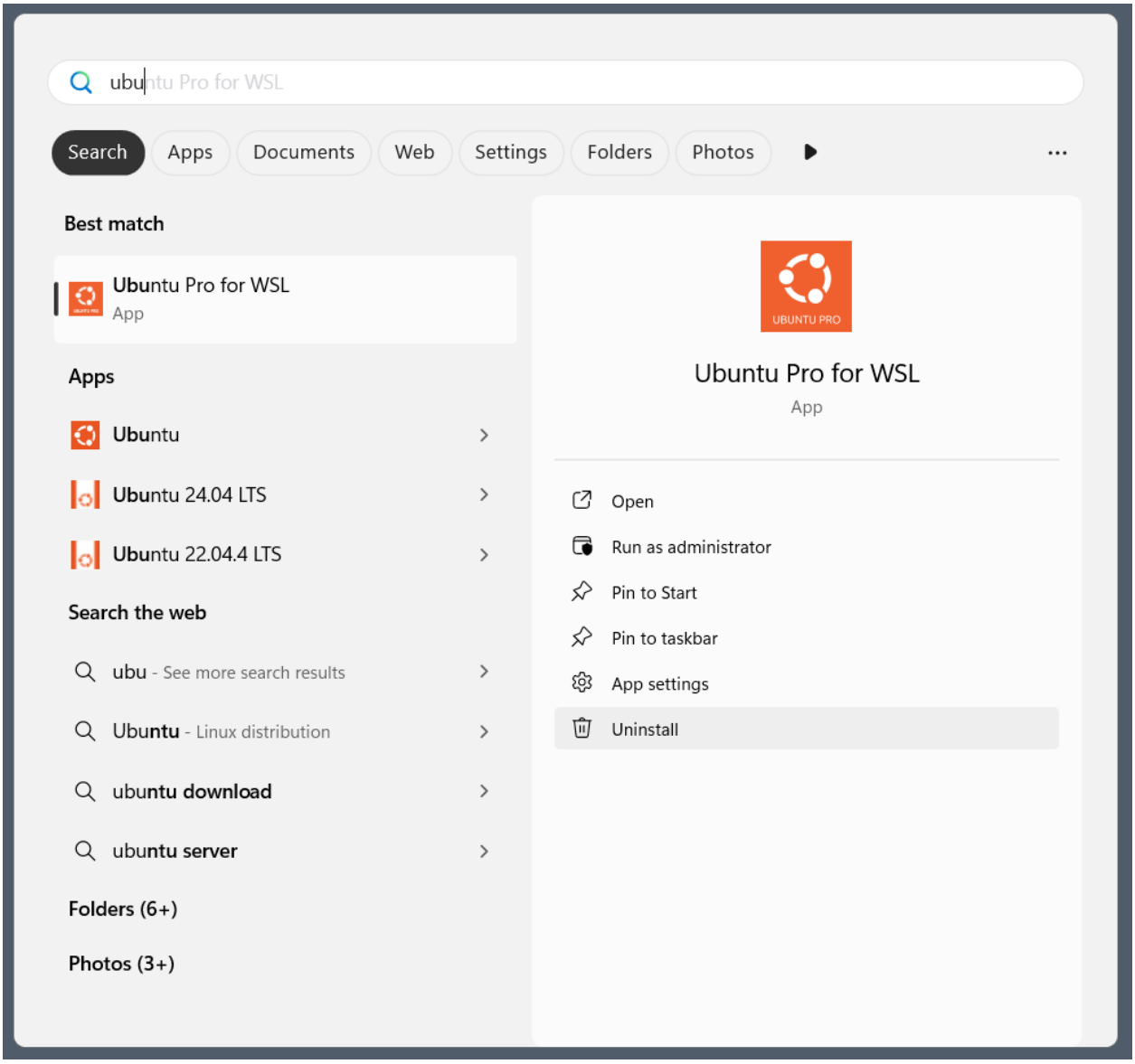
```
me@mib:~$ apt list --installed | grep -m 3 opencv

libopencv-calib3d4.5d/jammy,now 4.5.4+dfsg-9ubuntu4 amd64 [installed,automatic]
libopencv-contrib4.5d/jammy,now 4.5.4+dfsg-9ubuntu4 amd64 [installed,automatic]
libopencv-core4.5d/jammy,now 4.5.4+dfsg-9ubuntu4 amd64 [installed,automatic]
```

You know how to leverage UP4W and Landscape to efficiently manage your Ubuntu WSL instances at scale.

2.1.5 Tear things down

Uninstall UP4W

In the Windows Start Menu, locate the "Ubuntu Pro for WSL" application and right-click on it, then click **Uninstall**.



Additionally remove the `.ubuntupro` directory from your Windows user profile directory.

```
PS C:\Users\me\tutorial> Remove-Item -Recurse -Force C:\Users\me\.ubuntupro
```

Remove Ubuntu WSL apps

> **Warning**
>
> **If you already have them pre-installed:**
>
> Refer to the *backup instructions* to restore your pre-existing instances.
>
> Otherwise, proceed with the commands below.

In PowerShell run the following command to stop WSL:

```
PS C:\Users\me\tutorial> wsl --shutdown
```

Then, in the Windows Start Menu, locate the "Ubuntu 24.04 LTS" application, right-click on it, and select "Uninstall".

The instances will be removed automatically.

Remove WSL app

Only do this if you don't need WSL on this Windows machine following the tutorial.

In the Windows Start Menu locate the "WSL" application, right-click on it then select "Uninstall".

2.1.6 Next steps

This tutorial has introduced you to the amazing things that can be achieved with UP4W.

In the rest of the documentation you can find *how-to guides* for completing specific tasks, *reference* material describing key information relating to UP4W and dedicated *documentation for developers*.

2.2 How-to guides

These how-to guides cover key operations and processes in UP4W.

2.2.1 How to install and configure UP4W

1. Check that you meet the prerequisites

To install and configure UP4W you will need:

- A Windows host
- An Ubuntu Pro token
- Verify that the *firewall rules are correctly set up*

1. Visit the Ubuntu Pro page to get a subscription.

   See more: Ubuntu | Ubuntu Pro > Subscribe. If you choose the personal subscription option (`Myself`), the subscription is free for up to 5 machines.

2. Visit your Ubuntu Pro Dashboard to retrieve your subscription token.

   See more: Ubuntu | Ubuntu Pro > Dashboard

- (*Only if you want to use UP4W with Landscape:*) A UP4W-compatible Landscape server
  - i.e., Landscape beta

  See: Landscape | Quickstart deployment

Sure:

1. Set up an Ubuntu WSL to act as the server:
   1. Install a new Ubuntu WSL distro:

```
wsl --install Ubuntu-22.04
```

2. Find out the Windows host IP: In the WSL distro named *Ubuntu-22.04*, run:

   ```
   wslinfo --networking-mode
   ```

   - If it says `mirrored`, the relevant IP is `127.0.0.1`. Take note of this address.
   - Otherwise, run the command `ip route | grep ^default` and take note of the IP address that is printed.

3. Set up a Landscape Beta server:

   1. Start a shell in your *Ubuntu-22.04* distro.
   2. Install the Landscape (beta) following the steps in the Landscape Quickstart deployment with the following considerations:

      - Make sure you install the beta version.
      - Your FQDN is the address you took note of in the previous step.

      See more: Landscape | Quickstart deployment

4. Take note of the following addresses:

   - Hostagent API endpoint: `${WINDOWS_HOST_IP}:6554`
   - Message API endpoint: `${WINDOWS_HOST_IP}/message-system`
   - Ping API endpoint: `${WINDOWS_HOST_IP}/ping`

5. Open a `Ubuntu-22.04` terminal and keep it open during the rest of the guide.

   - This ensures this distro keeps running in the background. See also: Microsoft's FAQ.

2. Store the following file somewhere in your Windows system. Name it `landscape-client.conf`. Replace the variables in the file with the relevant values for your server.

```
[host]
url = ${HOSTAGENT_API_ENDPOINT}

[client]
url = ${MESSAGE_API_ENDPOINT}
ping_url = ${PING_API_ENDPOINT}
account_name = standalone
```

- (*Only for the verify step:*) One or more UP4W-compatible Ubuntu WSL instances
  - i.e., from an `Ubuntu`, `Ubuntu-Preview`, or `Ubuntu-22.04+` distro and with `wsl-pro-service` installed
    - note: with a freshly installed or updated `Ubuntu-Preview` or `Ubuntu-22.04+`, `wsl-pro-service` comes automatically pre-installed

It depends:

1. Open a shell into your instance and check its distro version:

   ```
   cat /etc/os-release
   ```

   If the distro is *not* `Ubuntu`, `Ubuntu-Preview`, or `Ubuntu-22.04+`: Your instance cannot be made UP4W-compatible. Please create a new one that is compatible. Otherwise:

2. Open a shell into your distro and check if package `wsl-pro-service` is installed:

```
pkg -s wsl-pro-service | grep Status
```

If the status is *not* `Status: install ok installed`: Install it by running: `sudo apt update && sudo apt install -y wsl-pro-service`.

> **Note**
>
> Here we assume you already have WSL installed. Run `wsl --version` to verify; if there you get an error because it is not there, install it: `wsl --install`.

> **Warning**
>
> Here we assume you do not have any existing `Ubuntu-Preview` or `Ubuntu-22.04+` instances or you have them but do not mind overwriting them.
>
> - To view your current registered instances, run `wsl --list --quiet`.
> - To export, delete, and re-import an instance, see `wsl --export`, `wsl --unregister`, and `wsl --import`.
>
>   See more: Microsoft | WSL Documentation.

On your Windows host, in the Microsoft Store, search for the `Ubuntu-Preview` or `Ubuntu-22.04+` distro app, click **Install / Update**, then **Open**. Follow the instructions to set up the instance. Note: The instance will come with `wsl-pro-service` pre-installed.

2. Install UP4W

On your Windows host, go to the Microsoft Store, search for *Ubuntu Pro for WSL* and click on the result. Find the *Install* button. Click. Done.

3. Configure UP4W for Ubuntu Pro and Landscape

See also: *Ubuntu Pro*, *Landscape*

There are two ways in which you can configure UP4W for Ubuntu Pro and Landscape – using the UP4W GUI or using the Windows Registry.

Using the GUI

> **Note**
>
> With this method you can only configure UP4W on a single Windows host at a time.

See also: *UP4W GUI*

1. Open the Windows menu, search for "Ubuntu Pro for WSL", click.
2. Input your Ubuntu Pro token:
   1. Click on **Already have a token?**.

2. Write your Ubuntu Pro token and click **Confirm**.

3. Input your Landscape configuration:

    1. Click on **Quick setup**.
    2. Write the **FQDN** of your server.
    3. Leave the **registration key** field empty.
    4. Click the **Continue** button.

Using the registry

> **Note**
>
> This method can be adapted to configure UP4W on multiple Windows hosts at a time.

See also: *Windows registry*

1. Press Win+R, type `regedit.exe`, and click OK.
2. Navigate the tree to `HKEY_CURRENT_USER\Software\Canonical\UbuntuPro`.

> **Note**
>
> This key will not exist until you've run UP4W at least once. Otherwise, you'll have to create the key and values yourself. See more: Microsoft Learn | Windows registry information for advanced users

3. Input your Ubuntu Pro token:
    - Right-click `UbuntuProToken` > Modify > Write the Ubuntu Pro token.
4. Input your Landscape configuration:
    - Right-click `LandscapeConfig` > Modify > Write the Landscape config.

        See more: *UP4W Landscape config reference*.

4. Verify that UP4W is working

If either verification step fails, wait for a few seconds and try again. This should not take longer than a minute.

1. Start the UP4W GUI and check that your subscription is active.
    - To open the GUI, search *Ubuntu Pro for WSL* in the Windows menu and click on it.
    - The GUI will explicitly say that you are subscribed.
2. Open any of the distros you want to manage and check that it is pro-attached with `pro status`.

    See also: *Ubuntu Pro client*

3. Open Landscape and check that the host and distro were registered.

    See more: Landscape | View WSL host machines and child computers

2.2.2 How to install Landscape server in a WSL instance

Motivation

While a Landscape server typically runs on external computers, it can also be set up on a WSL instance on a Windows machine.

This is especially useful if you want to test UP4W on a single Windows device. For example, the *tutorial* can be completed with Landscape server running in a Ubuntu WSL instance. The Landscape server can then be used to manage other WSL instances running UP4W and the Landscape client.

Guide

In PowerShell, `shutdown` WSL then install the Ubuntu 24.04 LTS instance with the `--root` option.

```
PS C:\Users\me\tutorial> wsl --shutdown

PS C:\Users\me\tutorial> ubuntu2404.exe install --root
```

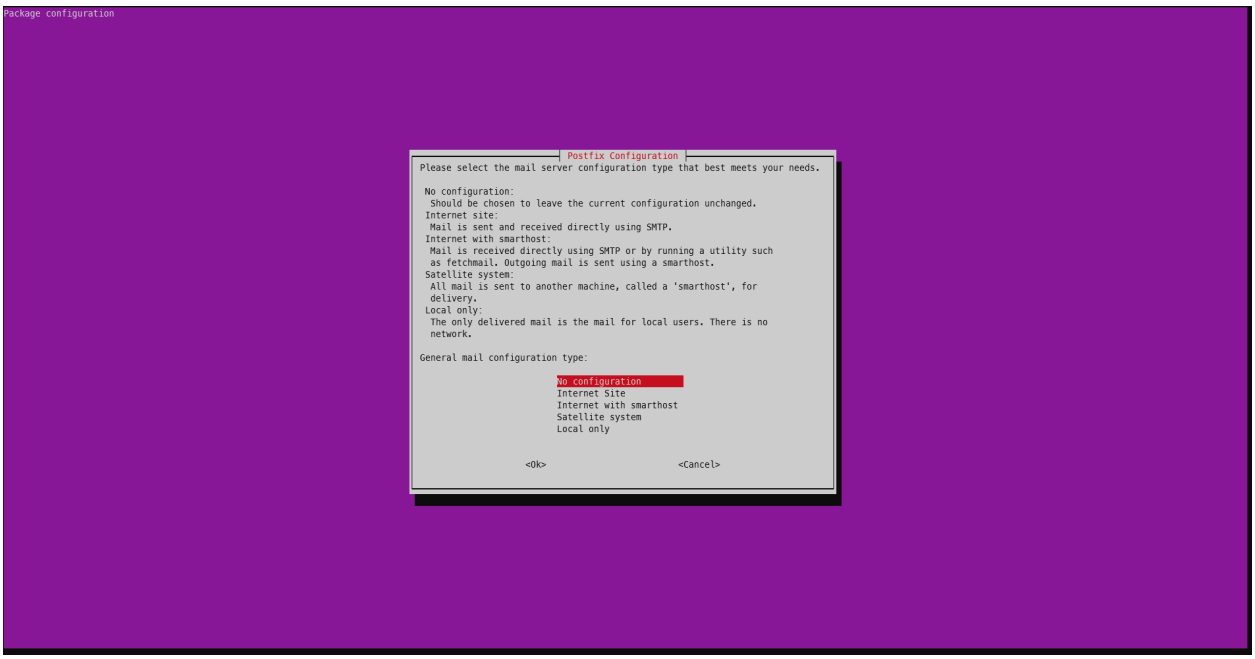
After successful installation log in to the new instance and add the landscape apt repository:

```
PS C:\Users\me\tutorial> ubuntu2204.exe

root@mib:~$ add-apt-repository ppa:landscape/self-hosted-beta -y
```

Update packages and then install the `landscape-server-quickstart` package.

```
root@mib:~$ apt update

root@mib:~$ apt install landscape-server-quickstart -y
```

A dialog will appear for 'Postfix configuration'. For 'General mail configuration type' select **No configuration**. Hit **Tab** to highlight the **Ok** button, press **Enter** and you will be returned to the shell prompt.

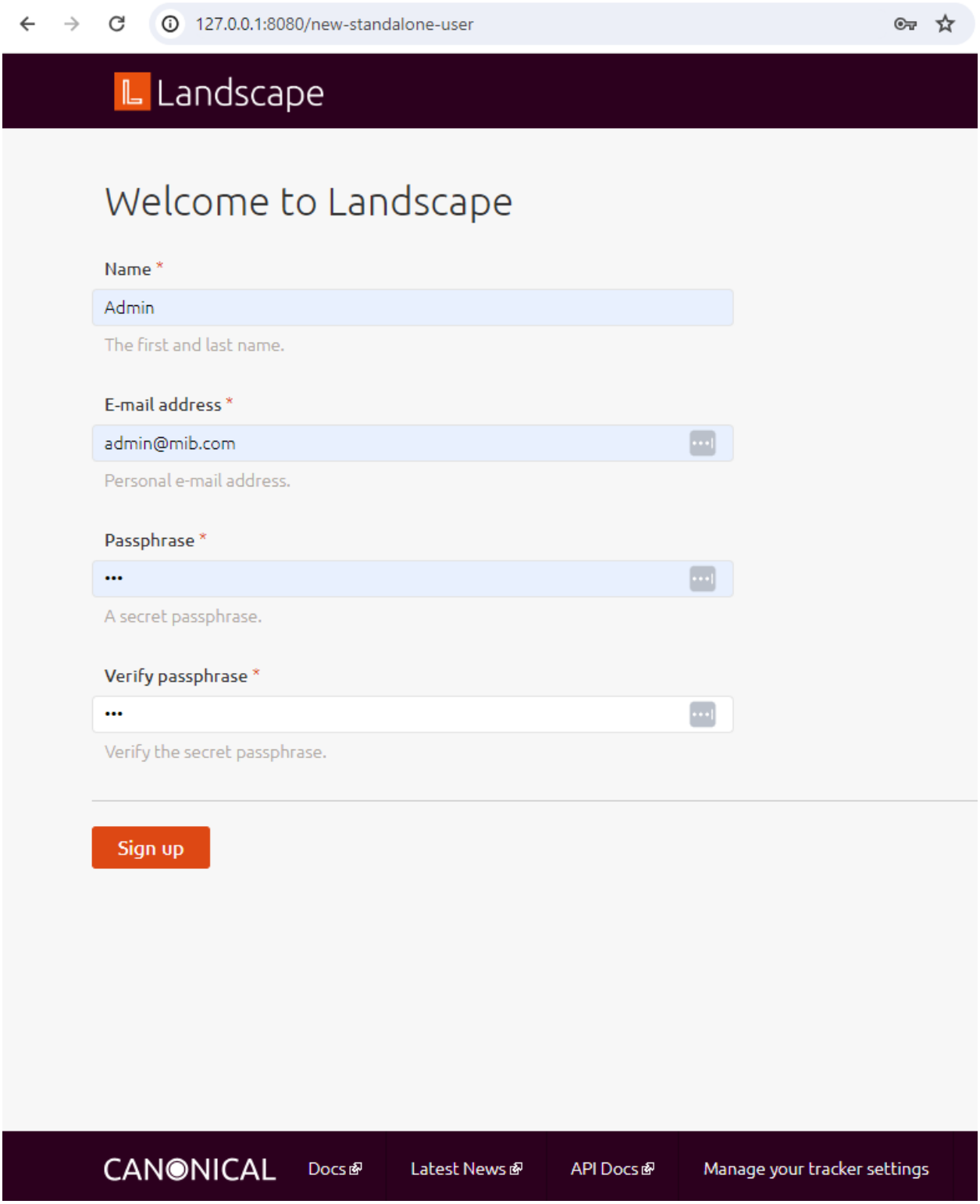


If Landscape has installed successfully, the log will indicate that Landscape systemd units are active. An example log is shown below for the first three units:

```
root@mib:~$ systemctl --state=running --no-legend --no-pager | grep -m 3 landscape
  landscape-api.service                 loaded active running LSB: Enable Landscape API
  landscape-appserver.service           loaded active running LSB: Enable Landscape␣
↪frontend UI
  landscape-async-frontend.service      loaded active running LSB: Enable Landscape␣
↪async frontend
```

Once installed Landscape will be served on `localhost` port 8080. Open your favourite browser on Windows and navigate to `http://127.0.0.1:8080` to create the Landscape global admin account. Enter the following credentials and click the **Sign Up** button:

Field	Value
Name	Admin
E-mail address	`admin@mib.com`
Passphrase	123
Verify passphrase	123



The Landscape client inside of any WSL instance will need the Landscape server certificate to connect to the server.

To achieve this copy the Landscape server certificate into your Windows user profile directory:

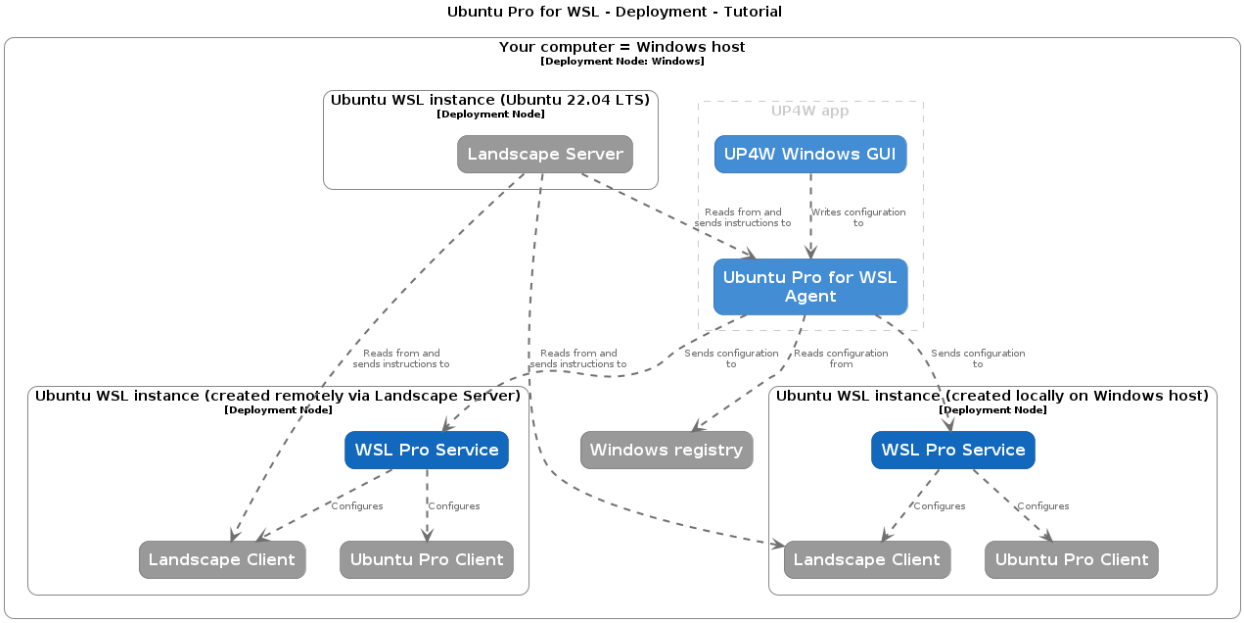
```
root@mib:~$ cp /etc/ssl/certs/landscape_server.pem /mnt/c/users/me/
```

Done – your self-hosted Landscape server is now up and running!

Now if you configure the Landscape client on any Ubuntu WSL instances to detect this server, they will also be registered with the Landscape service included in your Ubuntu Pro subscription.

The server will stay running until you close the terminal. If you do close the terminal running `ubuntu2404.exe` in a new terminal window will start the Landscape server automatically.

Using this server setup when following the *tutorial* would result in the following architecture:



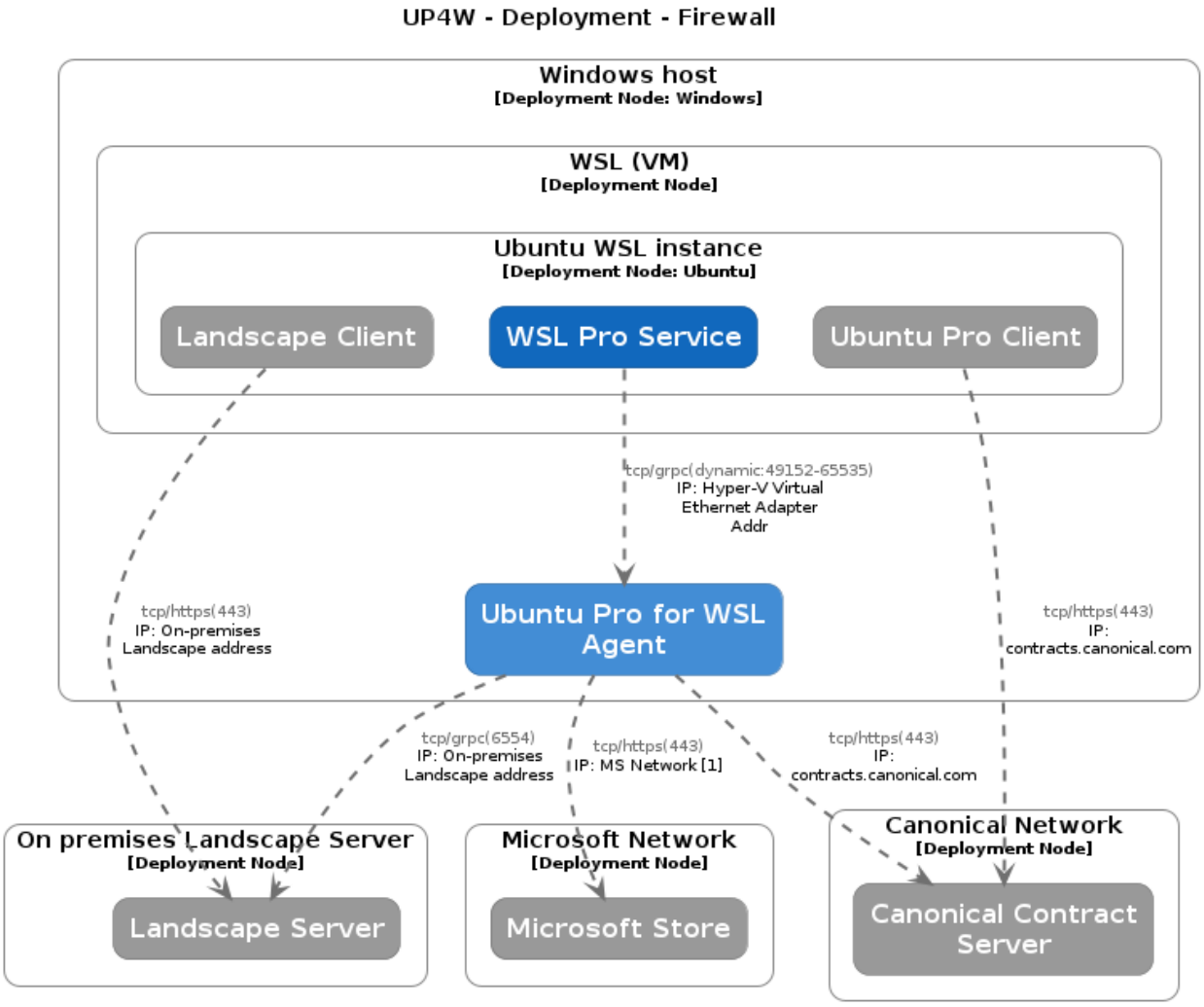
2.3 Reference

Our reference section contains more detailed information about the several pieces that make up the Ubuntu Pro for WSL tool and the value it provides.

2.3.1 Firewall requirements

Firewall rules must be configured for Ubuntu Pro for WSL to operate fully.

The following figure shows the possible connections between the different components and their default ports and protocols:



The following table lists the default ports and protocols used by Ubuntu Pro for WSL:

Description	Client System	Server System	Protocol	Default Port	Target address
Required for online installation of WSL instances[1].	Windows Host / Pro Agent	MS Store	tcp	https (443)	See Microsoft documentation for a list of addresses to allow.
Ubuntu Pro enablement[2]	Windows Host / Pro Agent	Canonical Contract Server	tcp	https (443)	`contracts.canonical.com`
Landscape management[2]	Windows Host / Pro Agent	Landscape Server	tcp	grpc (6554)	On-premise Landscape address
WSL instance management on the Windows host. Firewall rules set up at installation time of the WSL Pro agent.	WSL Instance / wsl-pro-service	Windows Host / Pro Agent	tcp	grpc (dynamic:49 65535)	Hyper-V Virtual Ethernet Adapter IP
Ubuntu Pro[2][3].	WSL Instance / Ubuntu Pro client	Canonical Contract Server	tcp	https (443)	`contracts.canonical.com`
Landscape[2].	WSL Instance / Ubuntu Pro client	Landscape Server	tcp	https (443)	On-premise Landscape address

If the client system is behind a proxy, ensure that the proxy is configured to allow the required connections.

2.3.2 Landscape

Landscape is a systems management tool designed to help you manage and monitor your Ubuntu systems from a unified platform.

> See more: Landscape | Documentation

In UP4W, Landscape consists of a remote server and two clients:

1. the usual Ubuntu-side client, in this case a *Landscape client* that comes automatically with any Ubuntu WSL instance, and
2. a Windows-side client, a Landscape client that is built into the *UP4W Windows Agent*.

The latter offers advantages unique to Ubuntu WSL – the ability to create new instances through Landscape and the ability to configure all your instances at scale (when you configure the Windows-side client, the UP4W agent forwards the configuration to the client on each instance).

[1] Access to the Microsoft Store is required for the online installation of WSL instances. Without it Ubuntu Pro for WSL will still be functional but it will not be possible to install WSL instances centrally from Landscape. In this case WSL instances have to be installed manually on the Windows hosts.

[2] Access to the contract server and Landscape server is required for proper operation of Ubuntu Pro for WSL.

[3] For air-gapped installation refer to the Ubuntu Pro documentation.

Landscape configuration schema

Both Landscape clients are configured via a single, plain text configuration file (e.g., `landscape.conf` or `landscape.ini`). This file is provided to the Windows host.

See more: *How to configure UP4W for Ubuntu Pro and Landscape*

The schema for this file is the same as Landscape for Ubuntu desktop or server, with a few additional keys specific to the WSL settings, which can be grouped into keys that affect just the Windows-side client and keys that affect both the Windows-side client and the Ubuntu WSL-side client(s). These additions are documented below.

See more: Landscape | Configure Ubuntu Pro for WSL for Landscape

Here is an example of what the configuration looks like:

```
[host]
url = landscape-server.domain.com:6554

[client]
url = https://landscape-server.domain.com/message-system
ping_url  = https://landscape-server.domain.com/ping
account_name = standalone
log_level = debug
ssl_public_key = C:\Users\user\Downloads\landscape_server.pem
```

Host

This section contains settings unique to the Windows-side client. Currently these consist of a single key:

- `url`: The URL of your Landscape account followed by a colon (`:`) and the port number. Port 6554 is the default for Landscape Quickstart installations.

Client

This section contains settings used by both clients. Most keys in this section behave the same way they would on a traditional Landscape setup. Only the following keys behave differently:

- `ssl_public_key`: This key must be a Windows path. The WSL instances will have this path translated automatically.
- `computer_title`: This key will be ignored. Instead, each WSL instance will use its Distro name as computer title.
- `hostagent_uid`: This key will be ignored.

See more: GitHub | Landscape client configuration schema

2.3.3 Landscape (client)

The Landscape client is a `systemd` unit running on *Landscape*-managed Ubuntu machines. It sends information about the system to the Landscape server. The server, in turn, sends instructions that the client executes.

In WSL, there is one Landscape client inside every Ubuntu WSL distro. The Landscape client comes pre-installed in your distro as part of the package `landscape-client`, but it must be configured before it can start running.

See more: Ubuntu manuals | Landscape client

UP4W will configure all Ubuntu WSL distros for you, so you don't need to configure each WSL instance separately; you specify the configuration once and UP4W will distribute it to every distro.

See more: *How to install and configure UP4W*

You can see the status of the Landscape client in any particular Ubuntu WSL instance by starting a shell in that instance and running:

```
systemctl status landscape-client.service
```

2.3.4 Ubuntu Pro

Ubuntu Pro is a subscription service offered by Canonical on top of the Long Term Support (LTS) releases of Ubuntu. It provides access to the following offerings:

- Landscape
- Center for Internet Security (CIS) compliance
- Expanded Security Maintenance (ESM)
- Canonical Support line

See more: Ubuntu | Ubuntu Pro

Ubuntu Pro services are managed by the *Ubuntu Pro client*.

Ubuntu Pro token

An Ubuntu pro token is a secret string of numbers and letters that acts as proof of purchase for your Ubuntu Pro subscription. Services provided by the Ubuntu Pro subscription will require a token to run.

See more: *Pro-attach*

You can find out what your Ubuntu Pro token is by visiting your Ubuntu Pro dashboard and logging in.

See more: Ubuntu | Ubuntu Pro dashboard

2.3.5 Ubuntu Pro (client)

The Ubuntu Pro client is a command-line utility (a CLI) that manages the different offerings of your Ubuntu Pro subscription. In UP4W, this executable is used within each of the managed WSL distros to enable *Ubuntu Pro* services within that distro.

This executable is provided as part of the `ubuntu-advantage-tools` package, which comes pre-installed in your Ubuntu WSL distros.

See more: Ubuntu manuals | Ubuntu advantage tools

Pro-attach

Pro-attaching a machine (e.g. a desktop computer, a WSL distro, a server, a virtual machine, etc.) means to provide your *Ubuntu Pro token* to the Ubuntu Pro client, so that it can enable Ubuntu Pro services.

2.3.6 Ubuntu WSL

Ubuntu WSL refers to the set of Ubuntu releases that target *WSL*. Each of these releases is distributed as a separate Microsoft Store application. Once installed, each of these releases will run as a separate WSL instance.

> See more: Ubuntu WSL | Documentation

2.3.7 Ubuntu Pro for WSL (UP4W)

Ubuntu Pro for WSL (UP4W) is an automation tool running on Windows hosts to manage Ubuntu WSL instances, providing them with compliance by attaching them to your Ubuntu Pro subscription and enrolling them into Landscape.

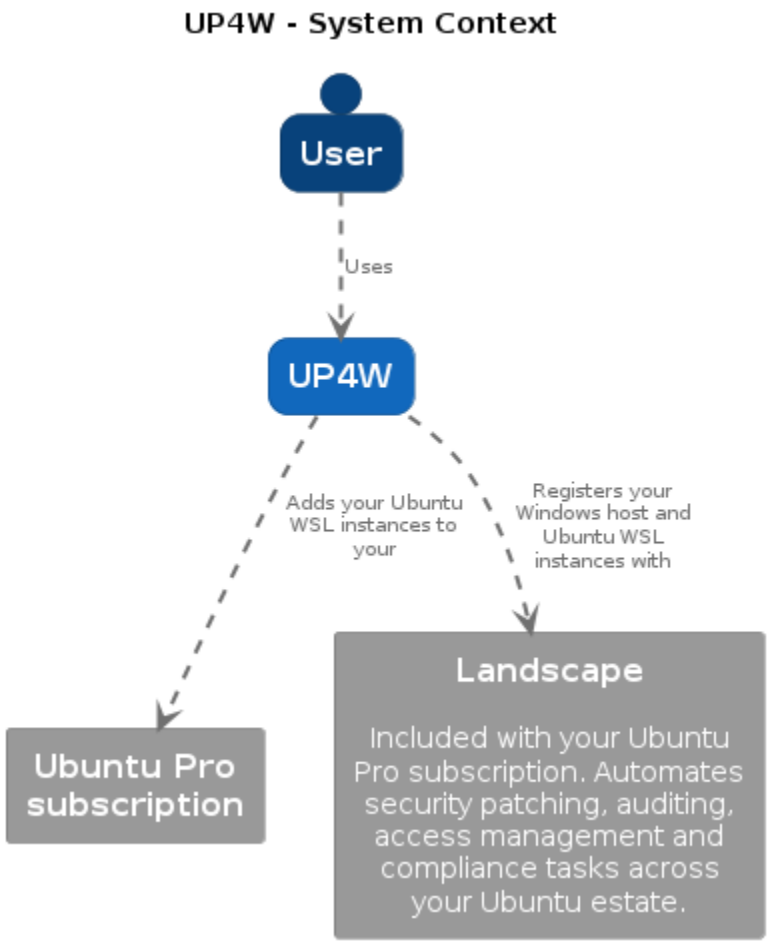
Some Ubuntu Pro for WSL components run on the Windows host:

- the *UP4W Windows Agent* providing automation services.
- the *UP4W GUI* for end users to manage their Ubuntu Pro subscription and Landscape configuration.

Additionally, Ubuntu Pro for WSL requires a component running inside each of the WSL distros:

- the *WSL Pro Service* communicates with the Windows Agent to provide automation services.

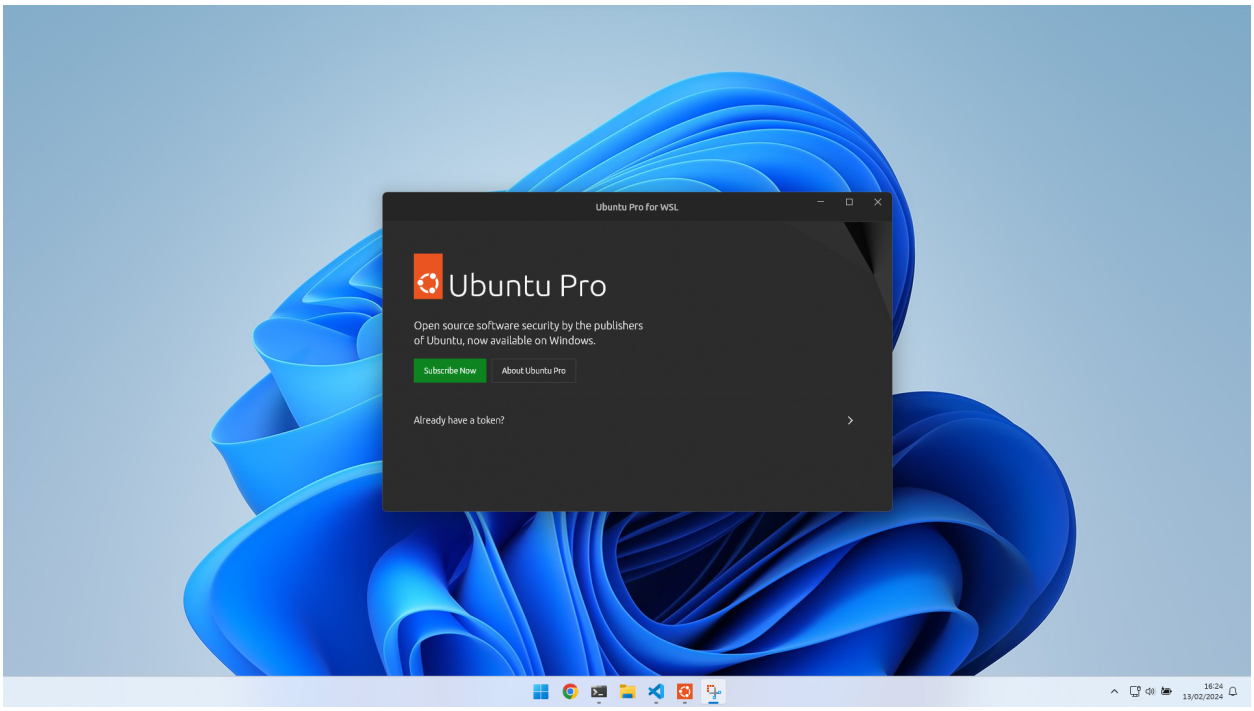
A top-level summary of the architecture is shown below:



2.3.8 UP4W - Graphical User Interface (GUI)

UP4W has a small GUI to help users with:

- Providing or acquiring an *Ubuntu Pro token*.
- Providing the *Landscape configuration*.



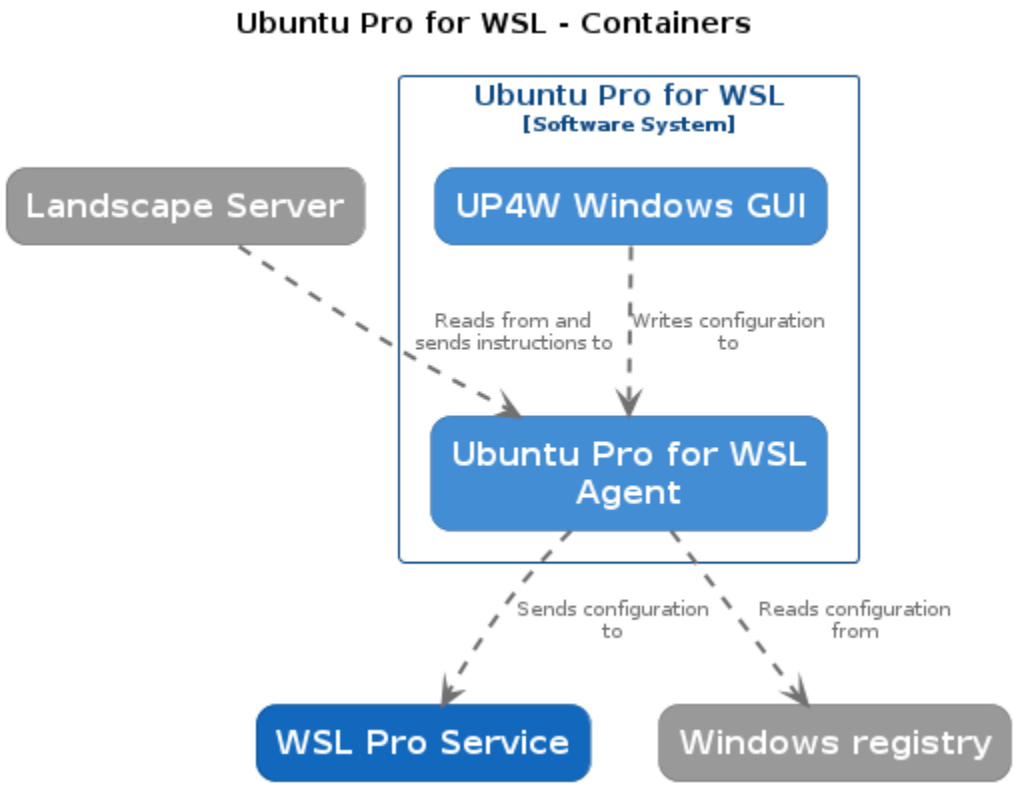
Interaction between the GUI and the agent

When the GUI starts, it attempts to establish a connection to the *UP4W Windows Agent*. If this fails, the agent is restarted. For troubleshooting purposes, you can restart the Agent by stopping the Windows process `ubuntu-pro-agent-launcher.exe` and starting the GUI.

2.3.9 UP4W - Windows Agent

UP4W's Windows Agent is a Windows application running in the background. It starts automatically when the user logs in to Windows. If it stops for some reason, it can also be started by launching the UP4W GUI.

The Windows agent is UP4W's central hub that communicates with all the components to coordinate them.



2.3.10 UP4W - WSL Pro service

A `systemd` unit running inside every Ubuntu WSL instance. The *Windows Agent* running on the Windows host sends commands that the WSL Pro Service executes, such as *pro-attaching* or configuring the *Landscape client*.

You can check the current status of the WSL Pro Service in any particular distro with:

```
systemctl status wsl-pro.service
```

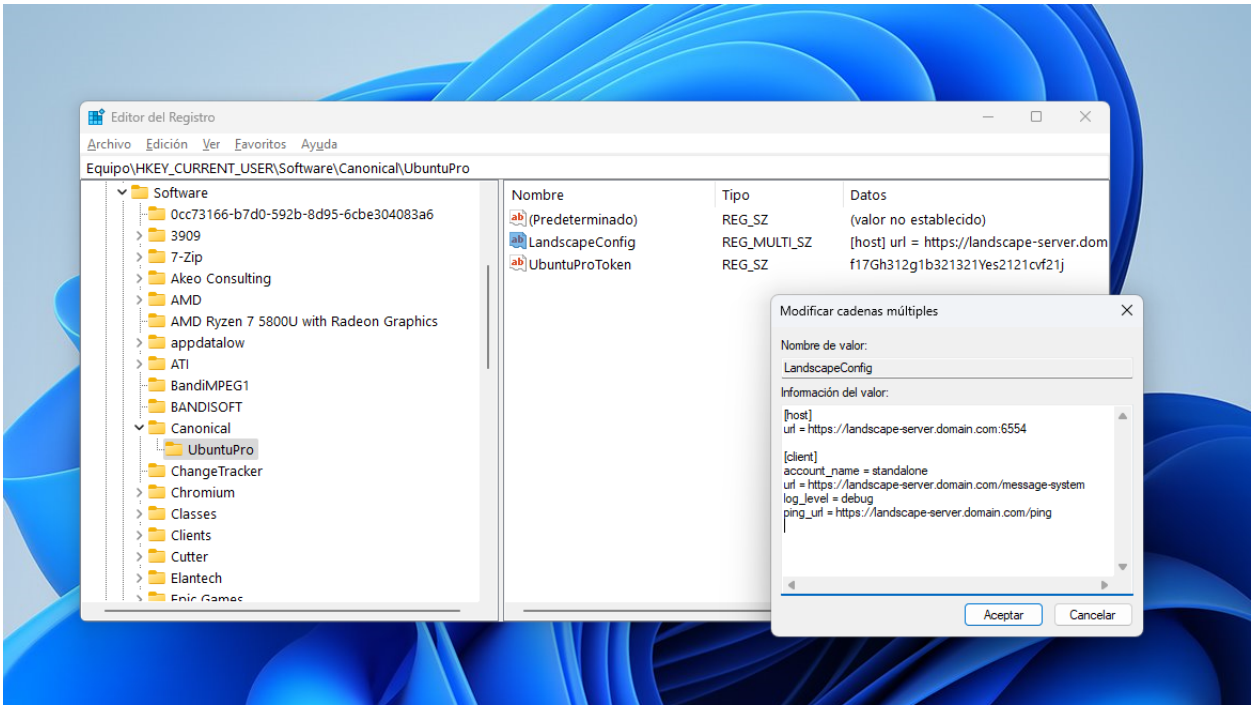
2.3.11 Windows registry

The Windows registry is a database provided by Windows where programs can read and write information. UP4W uses it as a read-only source of configuration.

> See more: Microsoft Learn | Windows registry information for advanced users

In UP4W, you can use the Windows registry to supply the configuration for *Ubuntu Pro* and *Landscape* to the *Windows Agent*.

> See more: *How to configure UP4W > Using the Windows registry*



Expected contents of the UbuntuPro registry key

The Windows agent will read the following values from the key at `HK_CURRENT_USER\Software\Canonical\UbuntuPro`:

- Value `UbuntuProToken` (type `String`) expects the *Ubuntu Pro token* for the user.
- Value `LandscapeConfig` (type `String` or `Multi-line string`) expects the *Landscape configuration*.

2.3.12 Windows Subsystem for Linux (WSL)

Windows Subsystem for Linux (WSL) is a Microsoft product, distributed as part of Windows 10 and Windows 11. It allows the user to run a Linux environment on the Windows machine, without the need for a traditional virtual machine or a dual boot setup.

See more: WSL documentation

2.4 Ubuntu Pro for WSL - developer documentation

Welcome

2.4.1 How-to guides

These how-to guides cover key operations and processes in Ubuntu Pro for WSL.

How to install UP4W

This guide will show you how to install UP4W for local development and testing.

Requirements:

- A Windows machine with access to the internet
- Appx from the Microsoft Store:
  - Windows Subsystem For Linux
  - Either Ubuntu, Ubuntu 22.04, or Ubuntu (Preview)
- The Windows Subsystem for Windows optional feature enabled

1. Download the Windows Agent and the WSL Pro Service

1. Go to the repository actions page.
2. Click the latest successful workflow run.
3. Scroll down past any warnings or errors, until you reach the Artifacts section.
4. Download:
   - Windows agent: UbuntuProForWSL+…-production
   - wsl-pro-service: Wsl-pro-service_…

Notice that, for the step above, there is also an alternative version of the MSIX bundle enabled for end-to-end testing. Most likely, that's not what you want to download.

2. Install the Windows Agent

This is the Windows-side agent that manages the distros.

1. Uninstall Ubuntu Pro for WSL if you had installed previously:

```
Get-AppxPackage -Name CanonicalGroupLimited.UbuntuPro | Remove-AppxPackage
```

2. Follow the download steps to download UbuntuProForWSL
3. Unzip the artefact
4. Find the certificate inside. Install it into `Local Machine/Trusted people`.
5. Double click on the MSIX bundle and complete the installation.
6. The Firewall may ask for an exception. Allow it.
7. The GUI should show up. You're done.

3. Install the WSL Pro Service

This is the Linux-side component that talks to the agent. Choose one or more distros Jammy or greater, and follow the instructions.

1. Uninstall the WSL-Pro-Service from your distro if you had it installed previously:

```
sudo apt remove wsl-pro-service
```

2. Follow the download steps to download the WSL-Pro-Service.
3. Unzip the artifact.
4. Navigate to the unzipped directory containing the .deb file. Here is a possible path:

```
cd /mnt/c/Users/WINDOWS-USER/Downloads/wsl-pro-service_*
```

5. Install the deb package.

```
sudo apt install ./wsl-pro-service_*.deb
```

6. Ensure it works via systemd:

```
systemctl status wsl-pro.service
```

How to restart UP4W

Some configuration changes only apply when you restart UP4W. Here is a guide on how to restart it. There are two options.

Option 1: Restart your UP4W host machine

This is the simple one. If you're not in a hurry to see the configuration updated, just wait until next time you boot your machine.

Option 2: Restart only UP4W

1. Stop the agent:

```
Get-Process -Name Ubuntu-Pro-Agent | Stop-Process
```

2. Stop the distro, or distros you installed WSL-Pro-Service in:

```
wsl --terminate DISTRO_NAME_1
wsl --terminate DISTRO_NAME_2
# etc.

# Alternatively, stop all distros:
wsl --shutdown
```

3. Start the agent again:
    1. Open the start Menu and search for "Ubuntu Pro for WSL".

2. The GUI should start.
3. Wait a minute.
4. Click on “Click to restart it”.

4. Start the distro, or distros you installed WSL-Pro-Service in.

How to access UP4W logs

At some point you may want to read the UP4W logs, most likely for debugging purposes. The agent and the service store their logs separately. This guide shows you where to find each of the logs.

Access the logs for the WSL Pro service

To access the logs of a specific distribution’s WSL-Pro-Service, you must first launch the distribution and then query the journal:

```
journalctl -u wsl-pro.service
```

For more information on using the journal, you can check out its man page with `man journalctl` or online.

These logs may be insufficient for proper debugging, so you may be interested in looking at the agent’s logs as well.

Access the logs for the Windows Agent

To access the logs for the Windows Agent:

1. Go to your home directory
   - Open the file explorer
   - Write `%USERPROFILE%` at the address
2. In the home directory, find the `.ubuntupro` directory and double-click on it.
3. In the `.ubuntupto` folder, find file `log` and open it with any text editor.
   - This file contains the logs sorted with the oldest entries at the top and the newest at the bottom.

How to reset UP4W back to factory settings

You can reset Ubuntu Pro for WSL to factory settings following these steps:

1. Shut down WSL

   ```
   wsl --shutdown
   ```

2. Uninstall the package and shut down WSL:

   ```
   Get-AppxPackage -Name "CanonicalGroupLimited.UbuntuPro" | Remove-AppxPackage`
   ```

3. Remove the public directory:

   ```
   Remove-Item -Recurse -Force "${env:UserProfile}\.ubuntupro\"
   ```

4. Remove the registry key:

1. Press Win+R
2. Type `regedit.exe` and click OK
3. Write `HKEY_CURRENT_USER\Software\Canonical\UbuntuPro` at the address bar
   - If this fails, you are done (the key does not exist).
4. Find the `UbuntuPro` key on the left
5. Right-click on it
6. Click delete

5. Install the Windows Agent package again (see the section on *how to install*). You do not need to re-install the WSL-Pro-Service.
6. You're done. Next time you start the GUI it'll be like a fresh install.

2.4.2 Reference

The reference material in this section provides technical descriptions of Ubuntu Pro for WSL.

Windows Agent CLI

See first: *UP4W - Windows Agent*

Usage

User commands

ubuntu-pro-agent

Ubuntu Pro for WSL agent

Synopsis

Ubuntu Pro for WSL agent for managing your pro-enabled distro.

```
ubuntu-pro-agent COMMAND [flags]
```

Options

```
  -c, --config string     configuration file path
  -h, --help              help for ubuntu-pro-agent
  -v, --verbosity count   issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

ubuntu-pro-agent clean

Removes all the agent's data and exits

```
ubuntu-pro-agent clean [flags]
```

Options

```
  -h, --help   help for clean
```

Options inherited from parent commands

```
  -c, --config string     configuration file path
  -v, --verbosity count   issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

ubuntu-pro-agent completion

Generate the autocompletion script for the specified shell

Synopsis

Generate the autocompletion script for ubuntu-pro-agent for the specified shell. See each sub-command's help for details on how to use the generated script.

Options

```
  -h, --help   help for completion
```

Options inherited from parent commands

```
  -c, --config string     configuration file path
  -v, --verbosity count   issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

ubuntu-pro-agent completion bash

Generate the autocompletion script for bash

Synopsis

Generate the autocompletion script for the bash shell.

This script depends on the 'bash-completion' package. If it is not installed already, you can install it via your OS's package manager.

To load completions in your current shell session:

```
source <(ubuntu-pro-agent completion bash)
```

To load completions for every new session, execute once:

Linux:

```
ubuntu-pro-agent completion bash > /etc/bash_completion.d/ubuntu-pro-agent
```

macOS:

```
ubuntu-pro-agent completion bash > $(brew --prefix)/etc/bash_completion.d/ubuntu-pro-
↪agent
```

You will need to start a new shell for this setup to take effect.

```
ubuntu-pro-agent completion bash
```

Options

```
  -h, --help              help for bash
      --no-descriptions   disable completion descriptions
```

Options inherited from parent commands

```
  -c, --config string     configuration file path
  -v, --verbosity count   issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

ubuntu-pro-agent completion fish

Generate the autocompletion script for fish

Synopsis

Generate the autocompletion script for the fish shell.

To load completions in your current shell session:

```
ubuntu-pro-agent completion fish | source
```

To load completions for every new session, execute once:

```
ubuntu-pro-agent completion fish > ~/.config/fish/completions/ubuntu-pro-agent.fish
```

You will need to start a new shell for this setup to take effect.

```
ubuntu-pro-agent completion fish [flags]
```

Options

```
  -h, --help              help for fish
      --no-descriptions   disable completion descriptions
```

Options inherited from parent commands

```
  -c, --config string     configuration file path
  -v, --verbosity count   issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

ubuntu-pro-agent completion powershell

Generate the autocompletion script for powershell

Synopsis

Generate the autocompletion script for powershell.

To load completions in your current shell session:

```
ubuntu-pro-agent completion powershell | Out-String | Invoke-Expression
```

To load completions for every new session, add the output of the above command to your powershell profile.

```
ubuntu-pro-agent completion powershell [flags]
```

Options

```
  -h, --help              help for powershell
      --no-descriptions   disable completion descriptions
```

Options inherited from parent commands

```
  -c, --config string     configuration file path
  -v, --verbosity count   issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

ubuntu-pro-agent completion zsh

Generate the autocompletion script for zsh

Synopsis

Generate the autocompletion script for the zsh shell.

If shell completion is not already enabled in your environment you will need to enable it. You can execute the following once:

```
echo "autoload -U compinit; compinit" >> ~/.zshrc
```

To load completions in your current shell session:

```
source <(ubuntu-pro-agent completion zsh)
```

To load completions for every new session, execute once:

Linux:

```
ubuntu-pro-agent completion zsh > "${fpath[1]}/_ubuntu-pro-agent"
```

macOS:

```
ubuntu-pro-agent completion zsh > $(brew --prefix)/share/zsh/site-functions/_ubuntu-pro-
↪agent
```

You will need to start a new shell for this setup to take effect.

```
ubuntu-pro-agent completion zsh [flags]
```

Options

```
  -h, --help              help for zsh
      --no-descriptions   disable completion descriptions
```

Options inherited from parent commands

```
  -c, --config string     configuration file path
  -v, --verbosity count   issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

ubuntu-pro-agent version

Returns version of agent and exits

```
ubuntu-pro-agent version [flags]
```

Options

```
  -h, --help   help for version
```

Options inherited from parent commands

```
  -c, --config string     configuration file path
  -v, --verbosity count   issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

Hidden commands

Those commands are hidden from help and should primarily be used by the system or for debugging.

WSL Pro Service CLI

See first: *UP4W - WSL Pro Service*

Usage

User commands

wsl-pro-service

WSL Pro Service

Synopsis

WSL Pro Service connects Ubuntu Pro for WSL agent to your distro.

```
wsl-pro-service COMMAND [flags]
```

Options

```
  -c, --config string     configuration file path
  -h, --help              help for wsl-pro-service
  -v, --verbosity count   issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

wsl-pro-service completion

Generate the autocompletion script for the specified shell

Synopsis

Generate the autocompletion script for wsl-pro-service for the specified shell. See each sub-command's help for details on how to use the generated script.

Options

```
  -h, --help   help for completion
```

Options inherited from parent commands

```
  -c, --config string     configuration file path
  -v, --verbosity count   issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

wsl-pro-service completion bash

Generate the autocompletion script for bash

Synopsis

Generate the autocompletion script for the bash shell.

This script depends on the 'bash-completion' package. If it is not installed already, you can install it via your OS's package manager.

To load completions in your current shell session:

```
source <(wsl-pro-service completion bash)
```

To load completions for every new session, execute once:

Linux:

```
wsl-pro-service completion bash > /etc/bash_completion.d/wsl-pro-service
```

macOS:

```
wsl-pro-service completion bash > $(brew --prefix)/etc/bash_completion.d/wsl-pro-service
```

You will need to start a new shell for this setup to take effect.

```
wsl-pro-service completion bash
```

Options

```
  -h, --help              help for bash
      --no-descriptions   disable completion descriptions
```

Options inherited from parent commands

```
  -c, --config string     configuration file path
  -v, --verbosity count   issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

wsl-pro-service completion fish

Generate the autocompletion script for fish

Synopsis

Generate the autocompletion script for the fish shell.

To load completions in your current shell session:

```
wsl-pro-service completion fish | source
```

To load completions for every new session, execute once:

```
wsl-pro-service completion fish > ~/.config/fish/completions/wsl-pro-service.fish
```

You will need to start a new shell for this setup to take effect.

```
wsl-pro-service completion fish [flags]
```

Options

```
  -h, --help              help for fish
      --no-descriptions   disable completion descriptions
```

Options inherited from parent commands

```
  -c, --config string     configuration file path
  -v, --verbosity count   issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

wsl-pro-service completion powershell

Generate the autocompletion script for powershell

Synopsis

Generate the autocompletion script for powershell.

To load completions in your current shell session:

```
wsl-pro-service completion powershell | Out-String | Invoke-Expression
```

To load completions for every new session, add the output of the above command to your powershell profile.

```
wsl-pro-service completion powershell [flags]
```

Options

```
  -h, --help              help for powershell
      --no-descriptions   disable completion descriptions
```

Options inherited from parent commands

```
  -c, --config string     configuration file path
  -v, --verbosity count   issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

wsl-pro-service completion zsh

Generate the autocompletion script for zsh

Synopsis

Generate the autocompletion script for the zsh shell.

If shell completion is not already enabled in your environment you will need to enable it. You can execute the following once:

```
echo "autoload -U compinit; compinit" >> ~/.zshrc
```

To load completions in your current shell session:

```
source <(wsl-pro-service completion zsh)
```

To load completions for every new session, execute once:

Linux:

```
wsl-pro-service completion zsh > "${fpath[1]}/_wsl-pro-service"
```

macOS:

```
wsl-pro-service completion zsh > $(brew --prefix)/share/zsh/site-functions/_wsl-pro-
↪service
```

You will need to start a new shell for this setup to take effect.

```
wsl-pro-service completion zsh [flags]
```

Options

```
  -h, --help              help for zsh
      --no-descriptions   disable completion descriptions
```

Options inherited from parent commands

```
  -c, --config string     configuration file path
  -v, --verbosity count   issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

wsl-pro-service version

Returns version of agent and exits

```
wsl-pro-service version [flags]
```

Options

```
  -h, --help   help for version
```

Options inherited from parent commands

```
  -c, --config string     configuration file path
  -v, --verbosity count   issue INFO (-v), DEBUG (-vv) or DEBUG with caller (-vvv) output
```

Hidden commands

Those commands are hidden from help and should primarily be used by the system or for debugging.

QA Process

Generalities

`wsl-pro-service` is seeded only on WSL images.

```
Build-dep: golang-go (\>= 2:1.21\~)
```

At any point in time, only the latest two versions of the Go toolchain receive security patches. Hence, we need to keep backporting new releases to fix vulnerabilities. They follow an approximate 6-month release cycle, so Go 1.21 should fall out of support by August 2024.

Process

WSL Pro Service follows a robust continuous integration and testing process. It is covered by a comprehensive automated test suite.

The team applies the following quality criteria:

- All changes are thoroughly reviewed and approved by core team members before integration.
- Each change is thoroughly tested at the unit, integration and system levels. All the tests pass in all supported architectures.
- Releases are reviewed as part of the SRU exception.

The test plan is **completely automated** and runs **every time a change is merged**, as well as **during packaging**. This covers integration and end-to-end tests. Integration tests run on each LTS affected by the SRU to ensure compatibility.

Testing also covers the upgrade from the current version to the proposed version.

Tests are not executed on different versions of Windows due to testing environment limitations.

Packaging QA

To prepare the release to LTS, the following procedure is being completed to ensure quality:

- All autopkgtests pass. Unit tests are executed as autopkgtests. Running higher-level tests would require a Windows VM. It is not available in autopkgtest at the moment. Even if wsl-pro-service tests could run in a VM, they wouldn't test anything real.
- The package does not break when upgrading.
- The binary is identical to the CI build, with only Debian packaging changes.
- The copyrights and changelog are up to date.
- An upgrade test from the previous package version has been performed using apt install/upgrade.

Code sanity

Code sanity checks are performed automatically on each build. They verify:

- Code linting.
- Go module files are up to date.
- Generated files are up to date.
- Any binary in the project builds.
- The Debian package builds.
- Vulnerabilities. It is a run of `govulncheck`.

All the layers are tested from APIs to mocks to the service itself

Example reports

- Code sanity and unit testing: QA workflow
- Integration tests: end-to-to-end tests workflow

Go Quality checks (ubuntu, wsl-pro-service) summary

Code sanity summary on /wsl-pro-service

Job	Status
Linting	🟢
Go module files up to date	🟢
Generated files up to date	🟢
Build	🟢
Vulnerability scanning	🟢

Job summary generated at run-time

Go Quality checks (ubuntu, common) summary

Code sanity summary on /common

Job	Status
Linting	🟢
Go module files up to date	🟢
Generated files up to date	🟢
Build	🟢
Vulnerability scanning	🟢

Job summary generated at run-time

Code coverage

There is no Codecov report due to the limitations of private projects. However, code coverage is calculated and displayed at testing time. Coverage is manually reviewed by the engineers.

Bug reporting

The main bug tracker remains on GitHub. GitHub Templates are available to help the user with the bug-reporting process and provide the right information.

Wsl-pro supports ubuntu-bug reporting to Launchpad with an apport hook but we are not collecting any data at the moment.

References

- Project Documentation
- Ubuntu Pro for WSL SRU exception
- Ubuntu Pro tools SRU exception