Incus 7.2 has been released

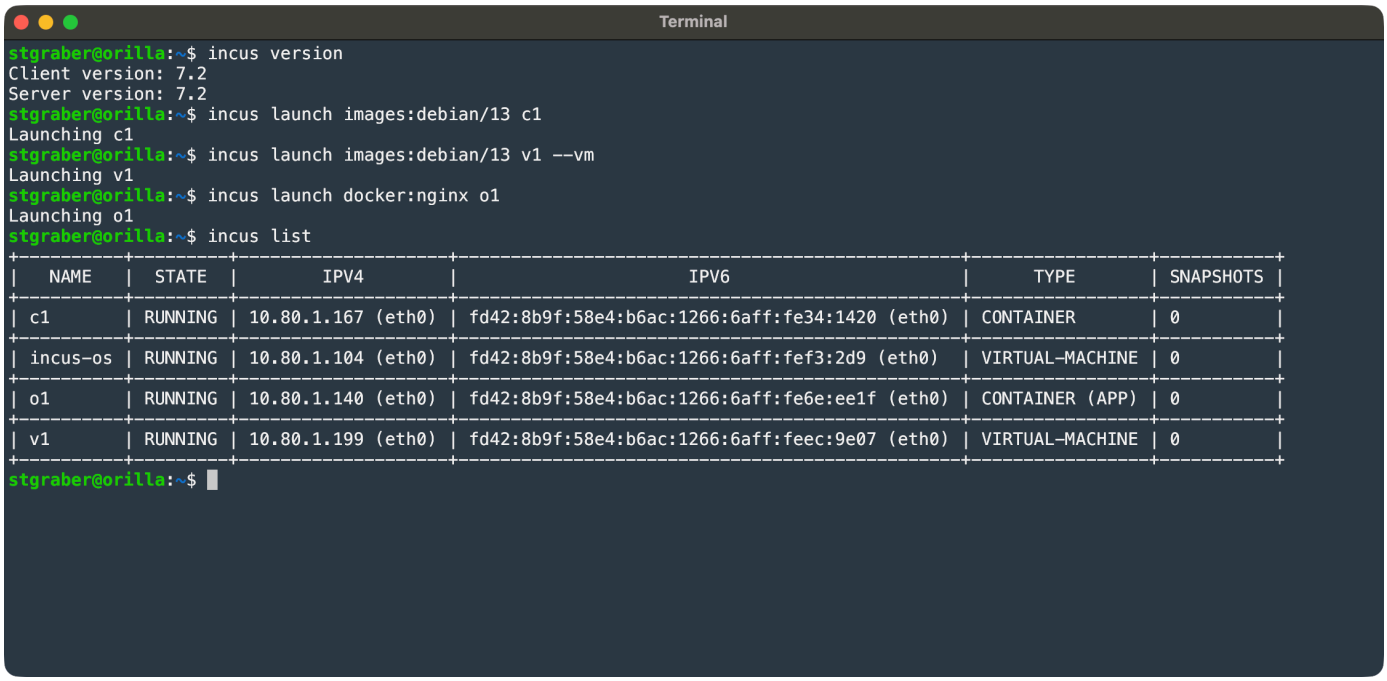
release, incus

---

stgraber (Stéphane Graber) 1 June 26, 2026, 7:35pm

The Incus team is pleased to announce the release of Incus 7.2!

It's another pretty busy release for us with a varied set of new features across the board as well as the usual set of performance improvements and bugfixes.



As usual, you can try Incus for yourself online: **Linux Containers - Incus - Try it online**

Security fixes

This release fixes 8 security issues:

- CVE-2026-48749 (critical) - Arbitrary file read+write on host via `rootfs/` symlink in malicious image
- CVE-2026-48750 (critical) - Arbitrary file write on host via `exec-output` symlink in crafted image
- CVE-2026-48751 (critical) - Restricted project bypass leading to arbitrary command execution
- CVE-2026-48752 (critical) - Arbitrary file read+write on host via `templates/` symlink in malicious image
- CVE-2026-48755 (critical) - Argument injection in backup compression algorithm leading to arbitrary file write and command execution
- CVE-2026-48769 (critical) - Arbitrary file write on client due to trusted image hash
- CVE-2026-55621 (high) - Project restriction bypass for custom volume copy across projects
- CVE-2026-55622 (high) - Project restriction bypass in instance copy across projects

New features

Per-instance SELinux integration

Incus now supports per-instance SELinux confinement for both containers and virtual machines, with automatic MCS (Multi-Category Security) level allocation to isolate instances from one another on the same host.

Four new instance configuration keys were introduced:

- `security.selinux.domain`: Override the SELinux process domain.
- `security.selinux.type`: Override the SELinux file type used for the instance storage.
- `security.selinux.level`: Override the SELinux MCS level.
- `security.selinux.label_rootfs`: Control rootfs labeling behavior (`auto`, `always` or `never`).

The computed context is persisted in the `volatile.selinux.context` key so that the allocated MCS range stays stable across restarts.

Documentation: **Instance options - Incus documentation**

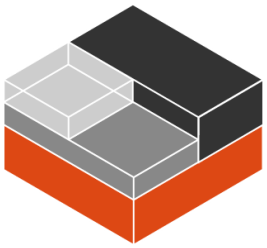
New `incus default` CLI command

A new `incus default` command has been added to help control CLI default options.

```
stgraber@orilla:~$ incus default show
list_format: ""
console_type: ""
console_spice_command: ""
no_color: false
stgraber@orilla:~$ incus default set list_format=compact
stgraber@orilla:~$ incus storage list
  NAME    DRIVER  DESCRIPTION  USED BY   STATE
 default  zfs                 11       CREATED
stgraber@orilla:~$ incus default unset list_format
stgraber@orilla:~$ incus storage list
+---------+--------+-------------+---------+---------+
|  NAME   | DRIVER | DESCRIPTION | USED BY |  STATE  |
+---------+--------+-------------+---------+---------+
| default | zfs    |             | 11      | CREATED |
+---------+--------+-------------+---------+---------+
stgraber@orilla:~$
```

The documentation for those options has also been updated:

> **CLI configuration file**



The Incus CLI client uses a configuration file to store environment data and modify its behavior. This documentation describes the file structure and its options. The configuration file: By default...

Filtered server info by default

`incus info` now defaults to a filtered view of the server information.

This hides (replaces with `SENSITIVE`) all private keys, certificates and other tokens by default, requiring `--show-sensitive` to reveal them. It also makes the output shorter by hiding the full list of API extensions and instead just showing a count.

Keepalive timeout from the CLI

The `incus remote` command gained a `set-keepalive` subcommand to configure (or disable) the connection keepalive timeout.

```
stgraber@orilla:~$ incus remote set-keepalive my-remote 30
stgraber@orilla:~$ incus remote set-keepalive my-remote 0
```

The CLI keepalive feature is used to maintain a connection with the remote server, making subsequent interactions a fair bit faster. This is particularly useful on higher latency connections as well as in environments where a lot of `incus` commands are being run.

Better OS-specific handling of CLI configuration

The CLI now stores its configuration and cache data in the correct per-OS location.

Up until now, the CLI tool would always use `~/.config/incus/` and `~/.cache/incus/` regardless of operating system, leading to an unusual location on MacOS and Windows.

With Incus 7.2, MacOS now uses `~/Library/Application Support/incus/` whereas Windows uses `%APPDATA%\incus`. The CLI will automatically relocate its configuration on first use.

Standalone server certificate update

A new `incus admin update-certificate` command allows replacing the server certificate on standalone (non-clustered) systems.

This is the equivalent to `incus cluster update-certificate` but for standalone systems. It replaces directly modifying the files in `/var/lib/incus/`.

Static network configuration for OCI containers

OCI application containers can now have their network statically configured.

The NIC `ipv4.address` and `ipv6.address` keys accept a CIDR value to statically configure the address inside the container, and the new `ipv4.gateway` and `ipv6.gateway` keys set the default gateway. Setting either address to `none` prevents any configuration for that address family and stops the built-in DHCP client from running on it.

For DNS, the new `oci.dns.nameservers`, `oci.dns.domain` and `oci.dns.search` instance configuration keys set the initial content of the container's `resolv.conf`, which is then extended with anything received over DHCP.

All of those keys are only valid for OCI containers.

```
stgraber@orilla:~$ incus create docker:nginx my-nginx
Creating my-nginx
stgraber@orilla:~$ incus config set my-nginx oci.dns.nameservers=1.0.0.1,1.1.1.1 
stgraber@orilla:~$ incus config device override my-nginx eth0 ipv4.address=10.10.1
Device eth0 overridden for my-nginx
stgraber@orilla:~$ incus start my-nginx
stgraber@orilla:~$ incus list my-nginx
+----------+---------+--------------------+-----------------------------------------
|   NAME   |  STATE  |       IPV4         |                     IPV6
+----------+---------+--------------------+-----------------------------------------
| my-nginx | RUNNING | 10.10.10.2 (eth0) | fd42:8b9f:58e4:b6ac:1266:6aff:fecb:e324
+----------+---------+--------------------+-----------------------------------------
```

Per-instance BGP route advertisement

Managed bridge networks gained two new configuration keys, `bgp.ipv4.instances` and `bgp.ipv6.instances`.

When enabled, Incus advertises a `/32` (IPv4) or `/128` (IPv6) route over BGP for each running instance connected to the network, withdrawing the route again when the instance stops. This makes it easy to route directly to individual instances in a BGP environment.

As part of that, Incus has now grown the ability to learn instance IP addresses by monitoring ARP/NDP for a limited period of time on instance startup.

Documentation: **Bridge network - Incus documentation**

Dynamic addresses in proxy NAT mode

Building on the new ability to detect instance IP addresses, proxy devices running in NAT mode can now use dynamic addresses and a wildcard listen address, removing the need to hardcode the instance address in

the proxy device configuration.

```
stgraber@orilla:~$ incus launch docker:nginx my-nginx
Launching my-nginx
stgraber@orilla:~$ incus config device add my-nginx http-80 proxy listen=tcp:0.0.0
Device http-80 added to my-nginx
stgraber@orilla:~$ incus list my-nginx
+----------+---------+---------------------+-----------------------------------
|   NAME   |  STATE  |        IPV4         |                      IPV6
+----------+---------+---------------------+-----------------------------------
| my-nginx | RUNNING | 10.80.1.162 (eth0) | fd42:8b9f:58e4:b6ac:1266:6aff:fe55:dbc
+----------+---------+---------------------+-----------------------------------
stgraber@orilla:~$ ip -4 a show dev enp0s5
2: enp0s5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP grou
    altname enx001c42e6a809
    inet 10.211.55.3/24 brd 10.211.55.255 scope global dynamic noprefixroute enp0s
       valid_lft 1031sec preferred_lft 1031sec
stgraber@orilla:~$ nc -v 10.211.55.3 1234
Connection to 10.211.55.3 1234 port [tcp/*] succeeded!
```

Expanded NBD access to VMs

A new `GET /1.0/instances/{name}/nbd` endpoint exposes all disks attached to a virtual machine over NBD, allowing concurrent access to every disk rather than a single volume at a time.

This is also wired up to a new `incus debug nbd` command which supports multiple concurrent client connections.

```
stgraber@orilla:~$ incus start v1
stgraber@orilla:~$ incus debug nbd v1
NBD listening on 127.0.0.1:36539
```

Those APIs are primarily meant for backup systems needing to easily and consistently access all of the drives on a VM. The NBD export also exposes the dirty bitmaps, allowing for easy incremental backups to be made.

Restoration can be done through the individual volume's NBD API when the instance is stopped.

Btrfs compression for storage volumes

A new `btrfs.compression` storage volume configuration key was added for the `btrfs` driver. It maps to the Btrfs `compression` property and takes the same values (for example `zstd`, `lzo`, `zlib` or `none`).

This can also be used to override compression settings from an existing btrfs filesystem, allowing for the Incus volumes to use a different compression algorithm as well as allowing turning off compression which then allows Incus to set the `nocow` flag on virtual machine disks.

Documentation: **Btrfs - btrfs - Incus documentation**

InfiniBand SR-IOV GUID configuration

`infiniband` devices using the `sriov` `nictype` now support two new configuration keys, `node_guid` and `port_guid`.

When set, the matching GUID of the allocated virtual function is changed to the provided value when the instance starts and restored to its original value when the instance stops.

Documentation: **Type: infiniband - Incus documentation**

Websocket origin restriction

A new `core.https_allowed_websocket_origin` server configuration key was added.

It can be set to a comma-separated list of allowed origins or to the `*` wildcard, controlling which origins are accepted on websocket connections.

This is useful for cross-origin access as well as in some proxied environments.

Documentation: **Server configuration - Incus documentation**

Deferred function logging

A large, repository-wide change in this release adds logging for deferred cleanup calls.

As a result, users may notice a number of new `WARNING` level log entries (for example around closing files, sockets or response bodies). This is expected and does not indicate a regression; these errors were always occurring but were simply discarded before.

If you notice spurious/repetitive such messages, please let us know so they can be investigated or silenced.

Complete changelog

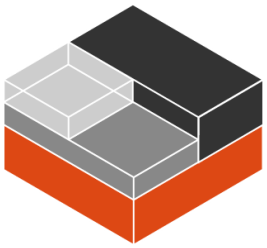
Here is a complete list of all changes in this release:

► Full commit list

Documentation

The Incus documentation can be found at:

> **Incus**



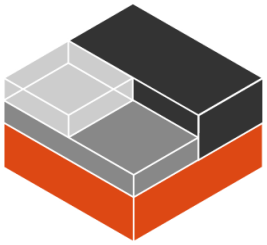
Incus is a modern, secure and powerful system container and virtual machine manager. It provides a unified experience for running and managing full Linux systems inside containers or virtual machin...

Packages

There are no official Incus packages as Incus upstream only releases regular release tarballs. Below are some available options to get Incus up and running.

Installing the Incus server on Linux

Incus is available for most common Linux distributions. You'll find detailed installation instructions in our documentation.



How to install Incus

The easiest way to install Incus is to install one of the available packages, but you can also install Incus from the sources. After installing Incus, make sure you have an incus-admin group on you...

Homebrew package for the Incus client

The client tool is available through HomeBrew for both Linux and MacOS.



incus

Homebrew's package index

Chocolatey package for the Incus client

The client tool is available through Chocolatey for Windows users.



Incus client 7.2.0

System container and virtual machine manager - client

Winget package for the Incus client

The client tool is also available through Winget for Windows users.



Install Incus with WinGet | winstall

Install Incus via WinGet. Copy the winget install command instantly. Incus is a modern, secure and powerful system container and virtual machine manager.

Support

Monthly feature releases are only supported up until the next release comes out. Users needing a longer support length and less frequent changes should consider using Incus 7.0 LTS instead.

Community support is provided at: **https://discuss.linuxcontainers.org**
Commercial support is available through: **Zabbly - Incus services**
Bugs can be reported at: **Issues · lxc/incus · GitHub**

11 Likes

---

jochumdev (René Jochum) 2 June 26, 2026, 8:19pm

Thanks!

---

turtle0x1 (Turtle0x1) 3 June 26, 2026, 8:55pm

Excited for NBD, be doing some low stakes exploring already but thought i'd ask - any "gotchas"/"you should know" before I get to deep?

---

stgraber (Stéphane Graber) 4 June 26, 2026, 9:18pm

It's VM-only and currently only provides persistent bitmap if on lvmcluster.
We've found a way to get persistent bitmaps on the other backends too though so that will come in Incus 7.3.

For containers, the recommended equivalent when talking backup systems is to use our SFTP API to map the container's filesystem, then perform an rsync-like comparison for the backup. It's a bit less fine grained than the NBD block approach but it's what's available with containers.

---

stgraber (Stéphane Graber) 5 June 26, 2026, 11:17pm

And the usual review video!

2 Likes

---

virtuous-sloth (Bruce Elrick) 7 June 27, 2026, 2:49pm

I know IncusOS generally ingests these Incus releases within a few days, but is there a easily-seen source showing either the first YYYYMMDDhhmm IncusOS that has each particular A.B release of Incus or the reverse, the Incus release of each and every IncusOS (outside of having it installed and interrogating it), so we know the mapping? Apologies for asking that here in the Incus forum but it seems like a relevant question after each Incus release.

---

gibmat (Mathias Gibbens) 8 June 27, 2026, 4:15pm

Probably the easiest way is to look at the automatically-generated changelogs of each IncusOS release. This won't directly tell you when the *first* release of IncusOS to include a specific version of Incus was, but will show you what versions have changed for each application and underlying Debian packages.

For example, looking at the most recent stable release **changelog** (202606270249):

```
current_version: "202606270249"
prior_version: "202606230053"
channel: stable
components:
  IncusOS:
    updated:
      - libssh2-1t64 version 1.11.1-1 to version 1.11.1-1+deb13u1
      - incus-osd version 202606230053 to version 202606270249
      - kpx version v1.12.3 to version v1.13.0
  incus:
    updated:
      - incus version 1:7.1-debian13-202605311451 to version 1:7.2-debian13-20260
```

```
      - incus-base version 1:7.1-debian13-202605311451 to version 1:7.2-debian13-2
      - incus-client version 1:7.1-debian13-202605311451 to version 1:7.2-debian13
      - incus-ui-canonical version 1:7.1-debian13-202605311451 to version 1:7.2-de
  incus-ceph:
    updated:
      - python3-urllib3 version 2.3.0-3+deb13u1 to version 2.3.0-3+deb13u2
  incus-lts-7.0:
    updated:
      - incus version 1:7.0-debian13-202605301824 to version 1:7.0-debian13-20260
      - incus-base version 1:7.0-debian13-202605301824 to version 1:7.0-debian13-2
      - incus-client version 1:7.0-debian13-202605301824 to version 1:7.0-debian13
      - incus-ui-canonical version 1:7.0-debian13-202605301824 to version 1:7.0-de
```

Hey, we got lucky and can see that this is the first IncusOS release with Incus 7.2. 😁 We also can see that security updates published by Debian for libssh2-1t64 and python3-urllib3 are included, along with a new version of kpx.

If you want to dig deeper and get specific versions of each package, you can look at each application's manifest.json file (for example, **Incus**). It's not quite a SBOM, but will show you exactly what's been installed during the build of IncusOS.

FYI, these specific links to IncusOS version 202606270249 are a bit ephemeral, as the public images server only keeps the past few IncusOS releases available. But each release follows the same pattern when publishing updates.

---

virtuous-sloth (Bruce Elrick) 9 June 27, 2026, 4:42pm

Thanks for the pointers, Mathias. I was looking at **https://images.linuxcontainers.org/os/** and figured the info must be somewhere in there.

As a career system admin I would expect such a mapping to be a well-surfaced fact about IncusOS, or, rather, I guess, each IncusOS application. I gather that the incus-os repo has some config that points to each of the Incus, OC, and MM repos that deterministically specify which commit is pulled in by the build system, so the facts can always be calculated during build, but I wonder if the tooling to calculate those facts could be used to record the actual versions back in the repository. Or maybe that assumption is wrong and the build system is just pulling in the latest commit, which is not deterministic?

Perhaps not by the build system but at least as part of the developer cycle.

I suppose it is a difference in mindset. I suspect developers think "well I can just calculate that fact, even in retrospect, by examining the various repository states at the build point in time", but for non-developers we just want to know the fact.

Thanks!

virtuous-sloth (Bruce Elrick) 10 June 27, 2026, 4:46pm

It just occurred to me that just as Incus itself has its LTS and monthly release tracks that eventually IncusOS will have parallel release tracks that track the included application release tracks, and that by that time the more-formal mapping I'd want would become necessary to record? /shrug/

---

virtuous-sloth (Bruce Elrick) 11 June 27, 2026, 4:50pm

Oh, I found `https://github.com/lxc/incus-os/incus-osd/go.mod` has the require `github.com/lxc/incus/v7 v7.2.0`, so perhaps what I'm looking for is the line in the require in that file that starts with `github.com/lxc/incus/`?

But I guess I would still need to correlate with the build artifacts.

---

stgraber (Stéphane Graber) 12 June 27, 2026, 6:23pm

The IncusOS repository pulls in the latest `stable` and `lts-7.0` packages from the Zabbly package repositories. It doesn't directly control the version of Incus that is included in its build.

What you found in go.mod is what version of some of the shared Incus code the `incus-osd` management daemon is using. But that's completely unrelated to the version of Incus included in a particular IncusOS image.

1 Like