# KubeCon | CloudNativeCon

# OPEN SOURCE SUMMIT

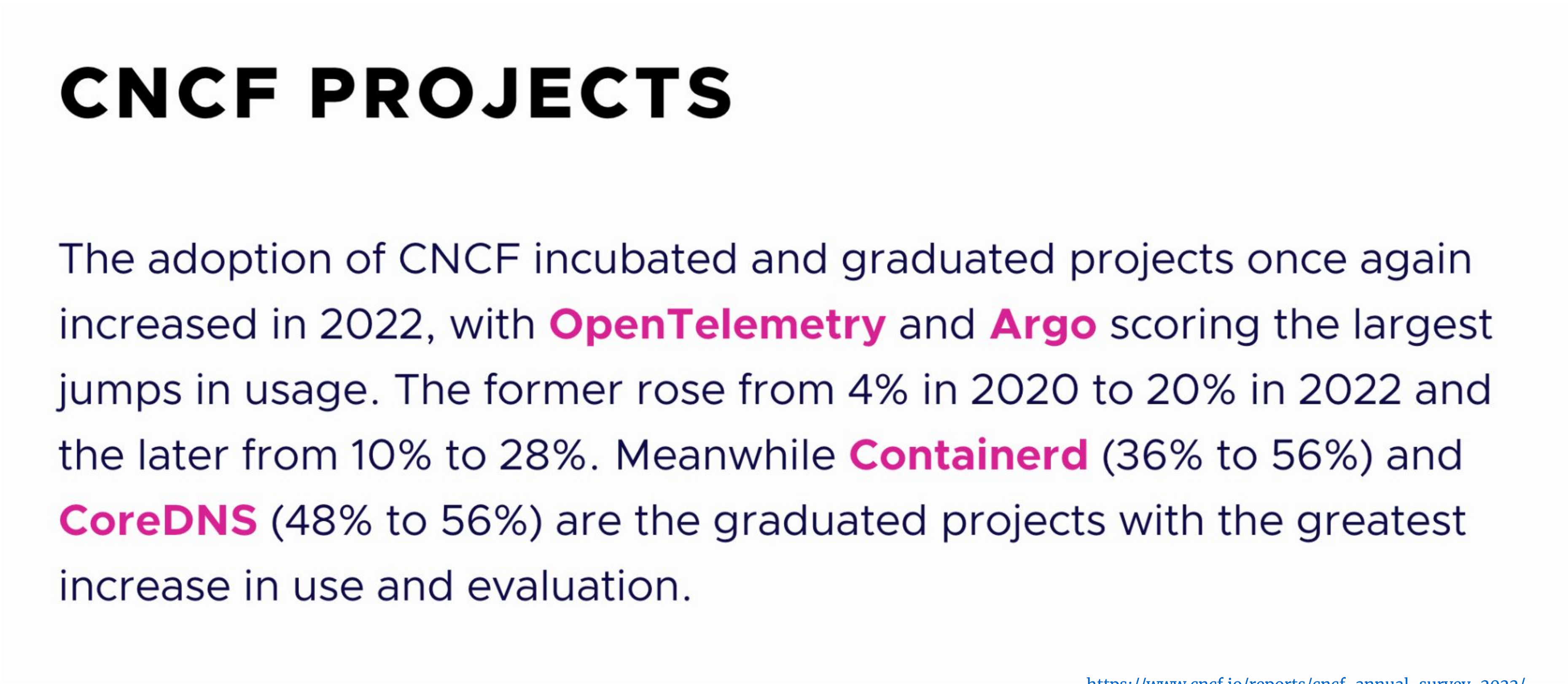
China 2023

KubeCon | CloudNativeCon
OPEN SOURCE SUMMIT
China 2023

# Project Update and Deep Dive: containerd

*Wei Fu, Microsoft*
*Iceber Gu(蔡威), DaoCloud*

# Community growth

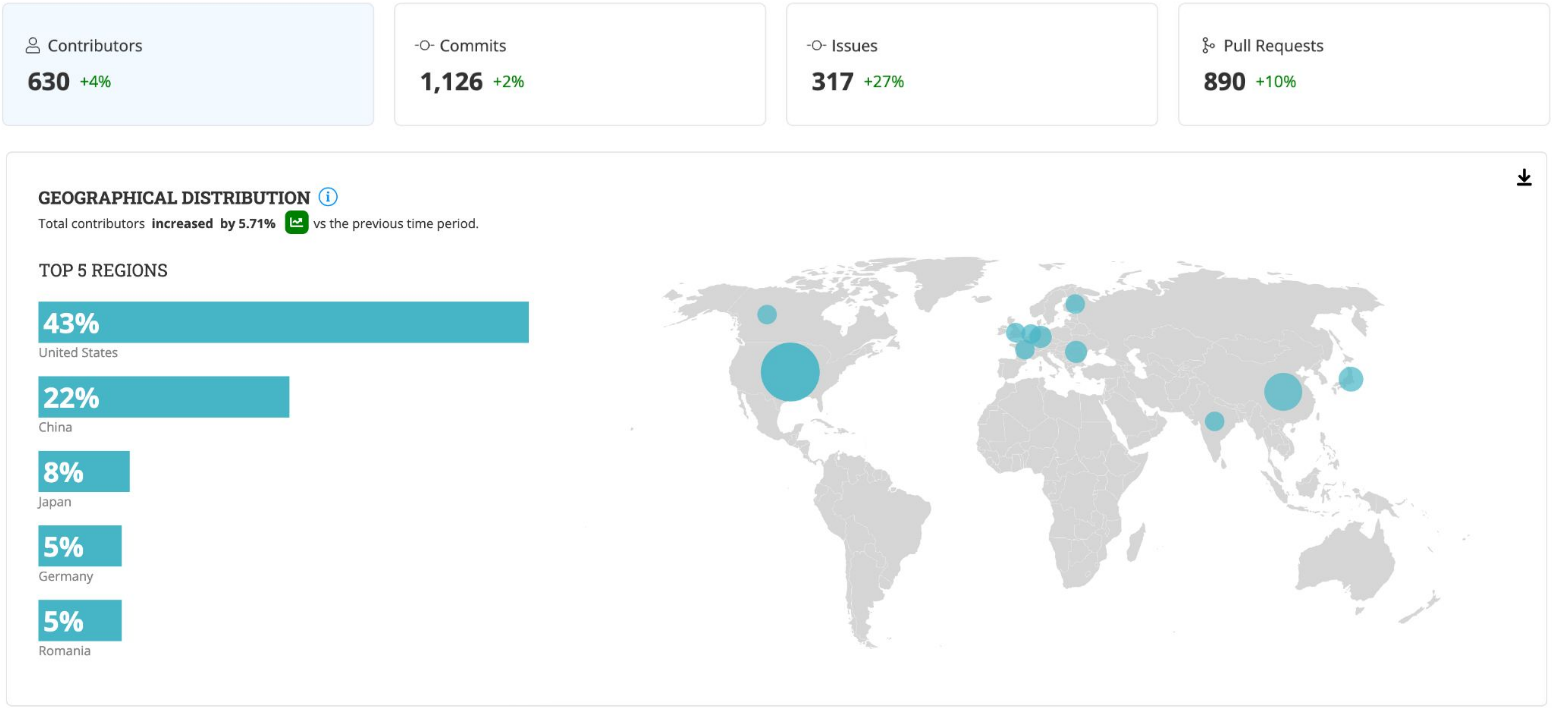


## CNCF PROJECTS

The adoption of CNCF incubated and graduated projects once again increased in 2022, with **OpenTelemetry** and **Argo** scoring the largest jumps in usage. The former rose from 4% in 2020 to 20% in 2022 and the later from 10% to 28%. Meanwhile **Containerd** (36% to 56%) and **CoreDNS** (48% to 56%) are the graduated projects with the greatest increase in use and evaluation.

https://www.cncf.io/reports/cncf-annual-survey-2022/

# Community growth



Contributors 630 +4%

Commits 1,126 +2%

Issues 317 +27%

Pull Requests 890 +10%

GEOGRAPHICAL DISTRIBUTION

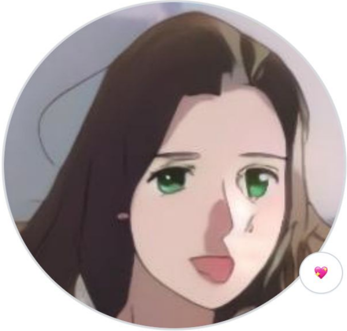
Total contributors **increased by 5.71%** vs the previous time period.

TOP 5 REGIONS

| | |
|---|---|
| 43% | United States |
| 22% | China |
| 8% | Japan |
| 5% | Germany |
| 5% | Romania |

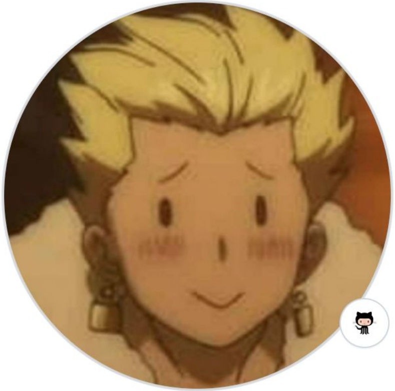
https://insights.v3.lfx.linuxfoundation.org/foundation/cncf/overview?project=containerd&bestPractice=false&repository=https:%2F%2Fgithub.com%2Fcontainerd%2Fcontainerd

# New maintainers



**Laura Brehm**
laurazard · she/her

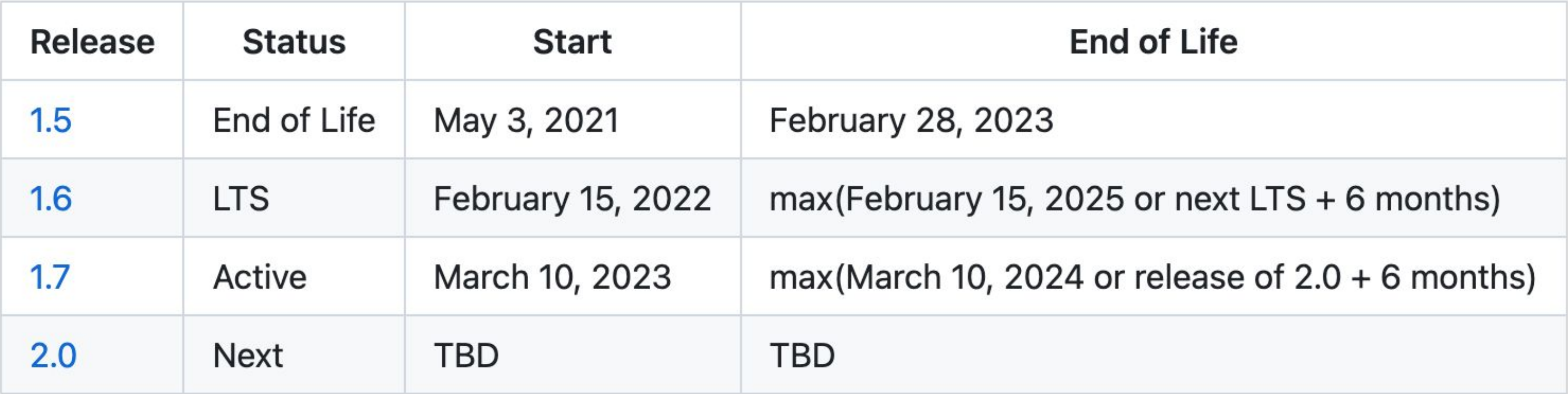
kiashok · she/her



**Iceber Gu**
Iceber

**Krisztian Litkey**
klihub

# Supported Releases



| Release | Status | Start | End of Life |
|---|---|---|---|
| 1.5 | End of Life | May 3, 2021 | February 28, 2023 |
| 1.6 | LTS | February 15, 2022 | max(February 15, 2025 or next LTS + 6 months) |
| 1.7 | Active | March 10, 2023 | max(March 10, 2024 or release of 2.0 + 6 months) |
| 2.0 | Next | TBD | TBD |

# containerd v1.6 - first LTS!

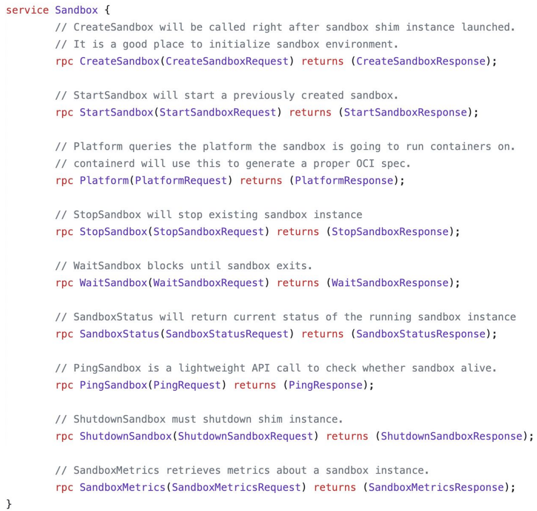
- Supported until **Feb 2025**
- Expand scope for backports
  - library dependency
  - toolchain (including Go)
  - compatibility with current Kubernetes versions
- Convert to a regular Active release with stricter backport criteria (Aug 2024)

# containerd v1.7 - last 1.x release

- **Sandbox Service and API** (New! - Experimental)
    - Shim-level API to support groups of containers
    - Preview CRI Plugin v2 - *ENABLE_CRI_SANDBOXES=1*
- **Node Resource Interface** (Updated - Experimental)
    - Extensions for OCI-compatible container runtimes
    - TTRPC
- **Transfer Service** (New! - Experimental)
    - Support to transfer artifact objects between any source and destination
- **User-Namespace Support** (New! - Experimental)
- **gRPC Shim Support** (New! - Experimental)

# Sandbox API

- New API to group container for shim
  - Sandbox API



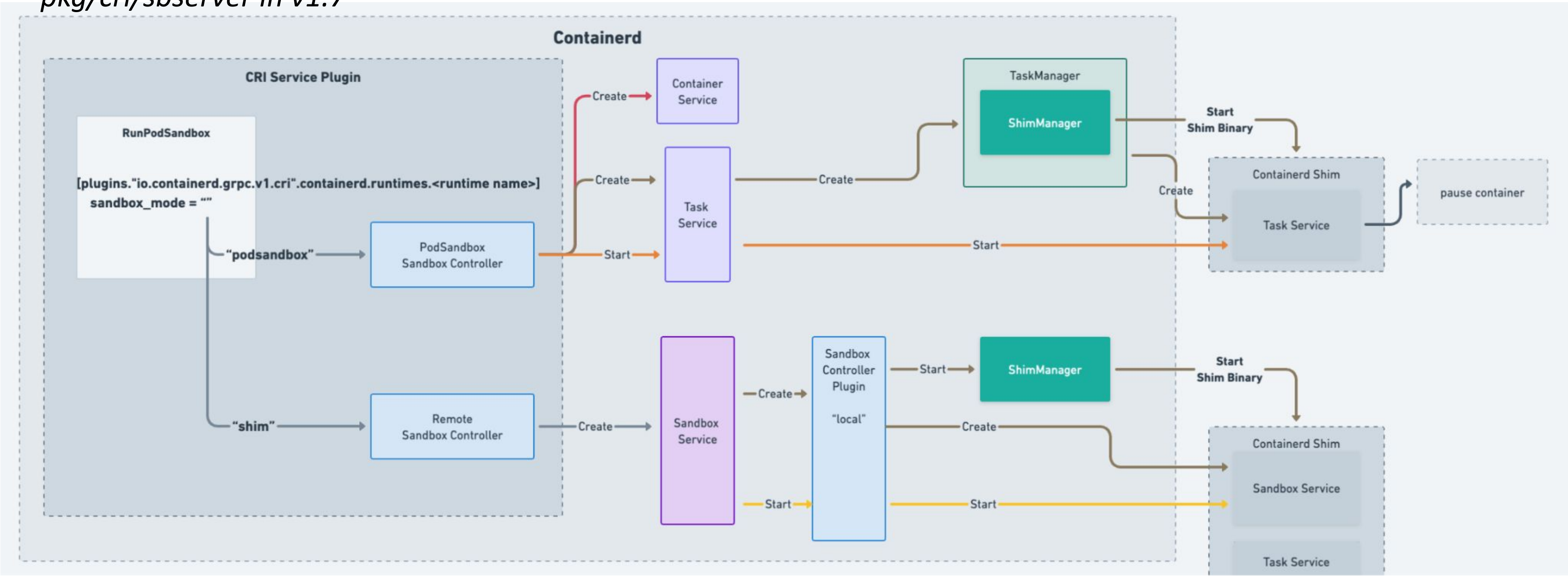
```
service Sandbox {
        // CreateSandbox will be called right after sandbox shim instance launched.
        // It is a good place to initialize sandbox environment.
        rpc CreateSandbox(CreateSandboxRequest) returns (CreateSandboxResponse);

        // StartSandbox will start a previously created sandbox.
        rpc StartSandbox(StartSandboxRequest) returns (StartSandboxResponse);

        // Platform queries the platform the sandbox is going to run containers on.
        // containerd will use this to generate a proper OCI spec.
        rpc Platform(PlatformRequest) returns (PlatformResponse);

        // StopSandbox will stop existing sandbox instance
        rpc StopSandbox(StopSandboxRequest) returns (StopSandboxResponse);

        // WaitSandbox blocks until sandbox exits.
        rpc WaitSandbox(WaitSandboxRequest) returns (WaitSandboxResponse);

        // SandboxStatus will return current status of the running sandbox instance
        rpc SandboxStatus(SandboxStatusRequest) returns (SandboxStatusResponse);

        // PingSandbox is a lightweight API call to check whether sandbox alive.
        rpc PingSandbox(PingRequest) returns (PingResponse);

        // ShutdownSandbox must shutdown shim instance.
        rpc ShutdownSandbox(ShutdownSandboxRequest) returns (ShutdownSandboxResponse);

        // SandboxMetrics retrieves metrics about a sandbox instance.
        rpc SandboxMetrics(SandboxMetricsRequest) returns (SandboxMetricsResponse);
}
```

- Sandbox Service

- Sandbox Controller interface
  - Handle sandbox environment for grouped containers
  - Support to manage multiple runtime platforms
    - Linux/Unix/Windows
    - Container, VM, microVM

# Sandbox API: Controller and Service

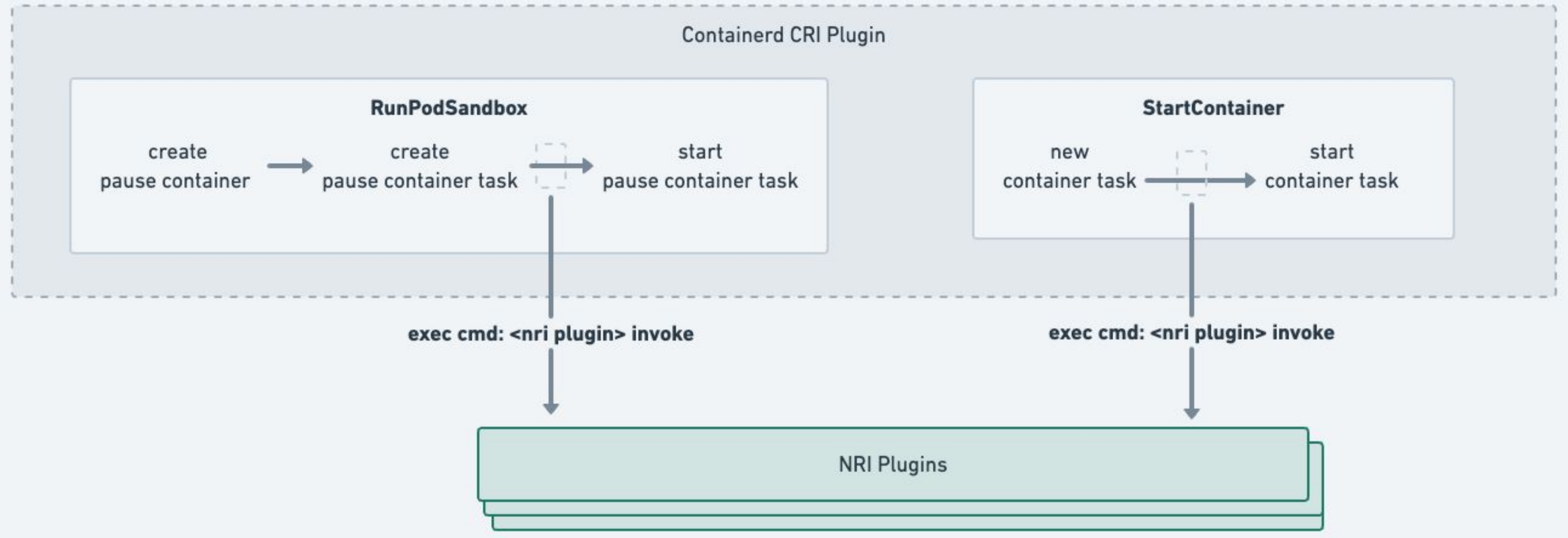
*pkg/cri/sbserver in v1.7*



- ENABLE_CRI_SANDBOXES=1 in v1.7
- Default in v2.0

# Node Resource Interface

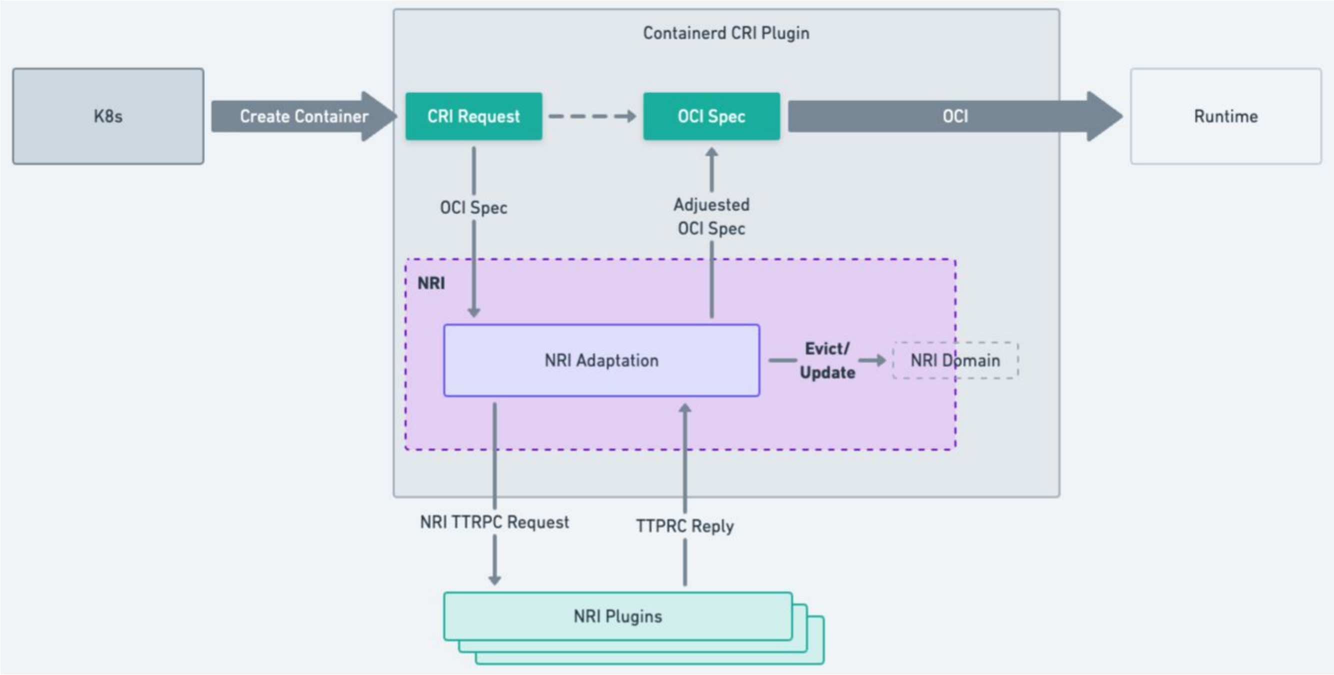
NRI v0.1: start the plugin binary



# Node Resource Interface

- Middleware extension between CRI and OCI
- ttRPC bindings

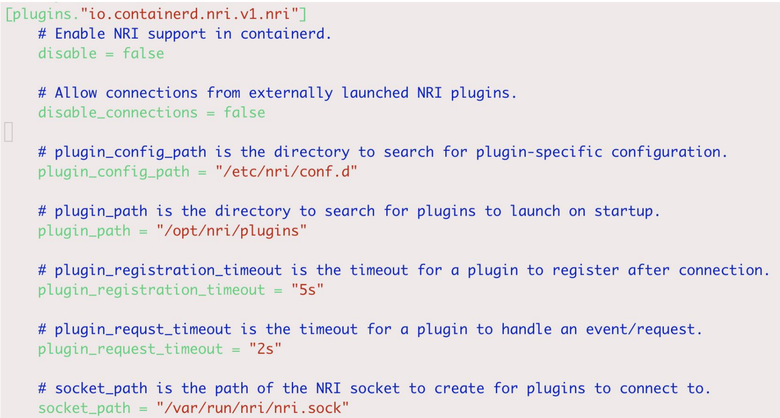
Create a Container



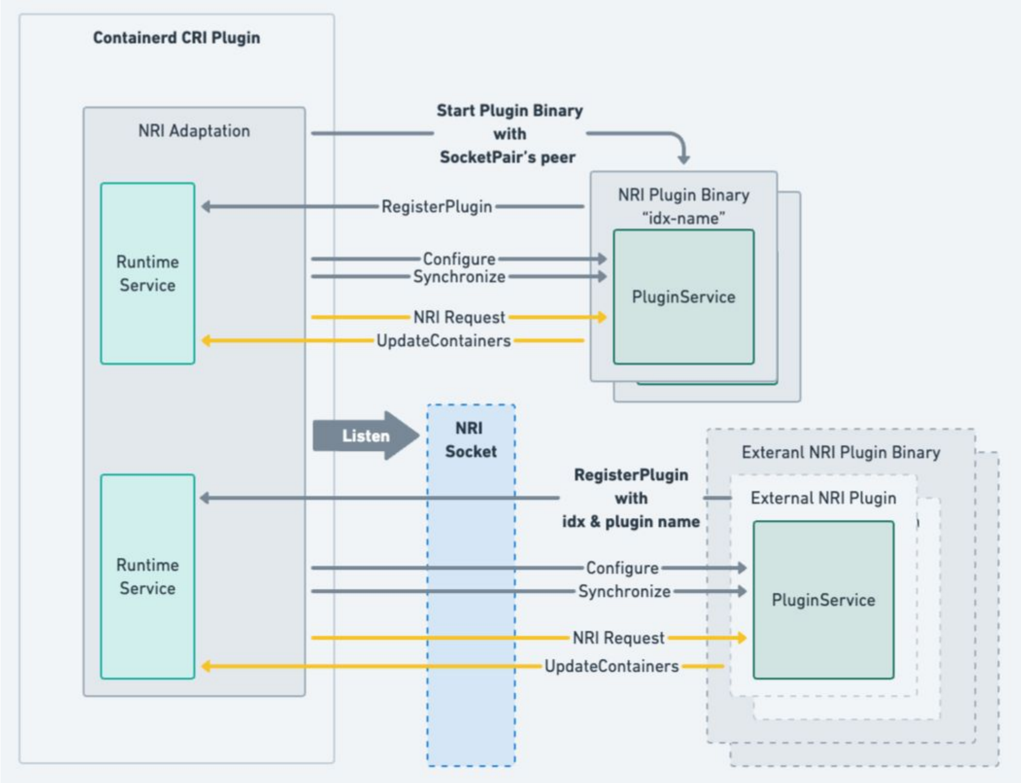
# Node Resource Interface

## Plugin Registration

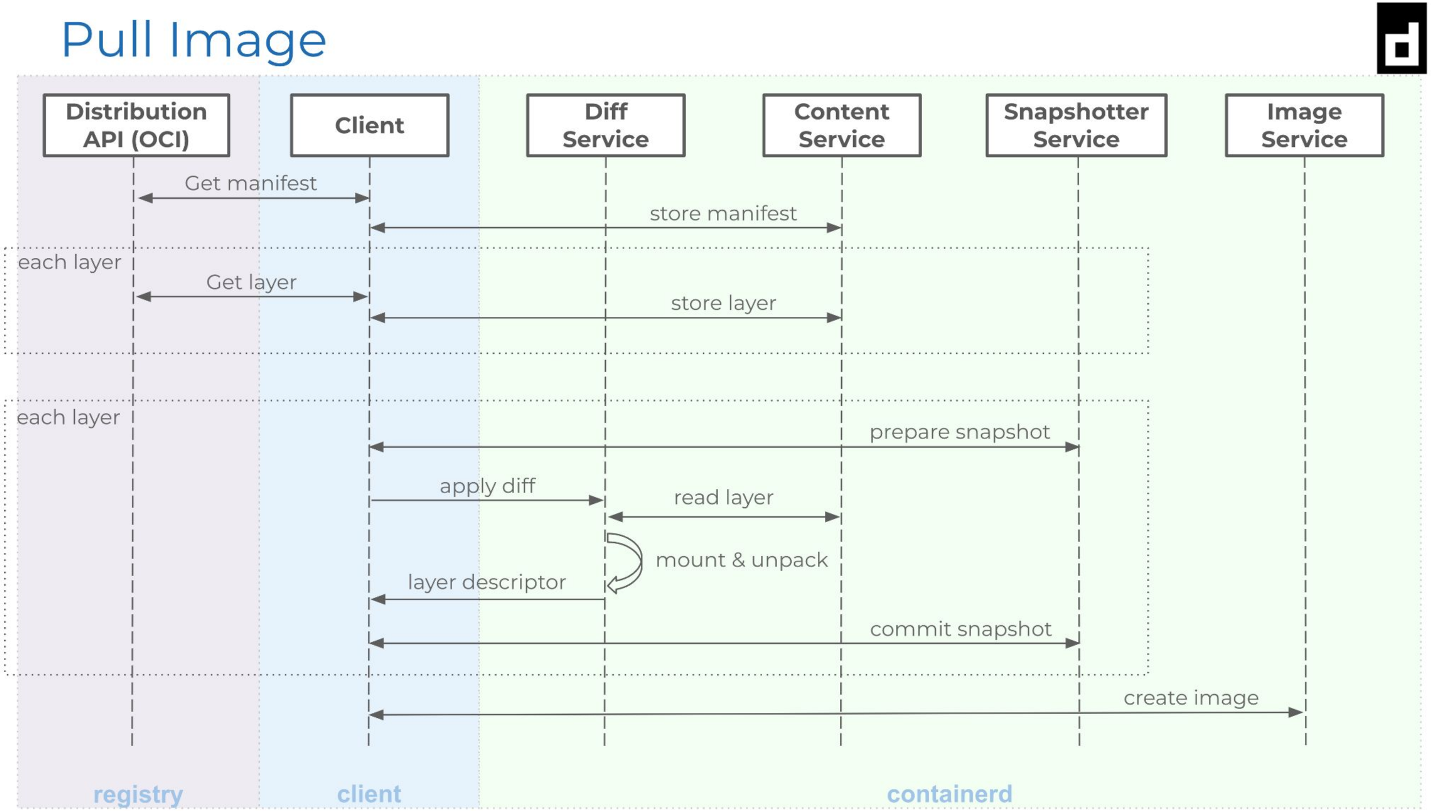
- NRI Plugin Binary
- External NRI Plugin



```
[plugins."io.containerd.nri.v1.nri"]
    # Enable NRI support in containerd.
    disable = false

    # Allow connections from externally launched NRI plugins.
    disable_connections = false

    # plugin_config_path is the directory to search for plugin-specific configuration.
    plugin_config_path = "/etc/nri/conf.d"

    # plugin_path is the directory to search for plugins to launch on startup.
    plugin_path = "/opt/nri/plugins"

    # plugin_registration_timeout is the timeout for a plugin to register after connection.
    plugin_registration_timeout = "5s"

    # plugin_requst_timeout is the timeout for a plugin to handle an event/request.
    plugin_request_timeout = "2s"

    # socket_path is the path of the NRI socket to create for plugins to connect to.
    socket_path = "/var/run/nri/nri.sock"
```

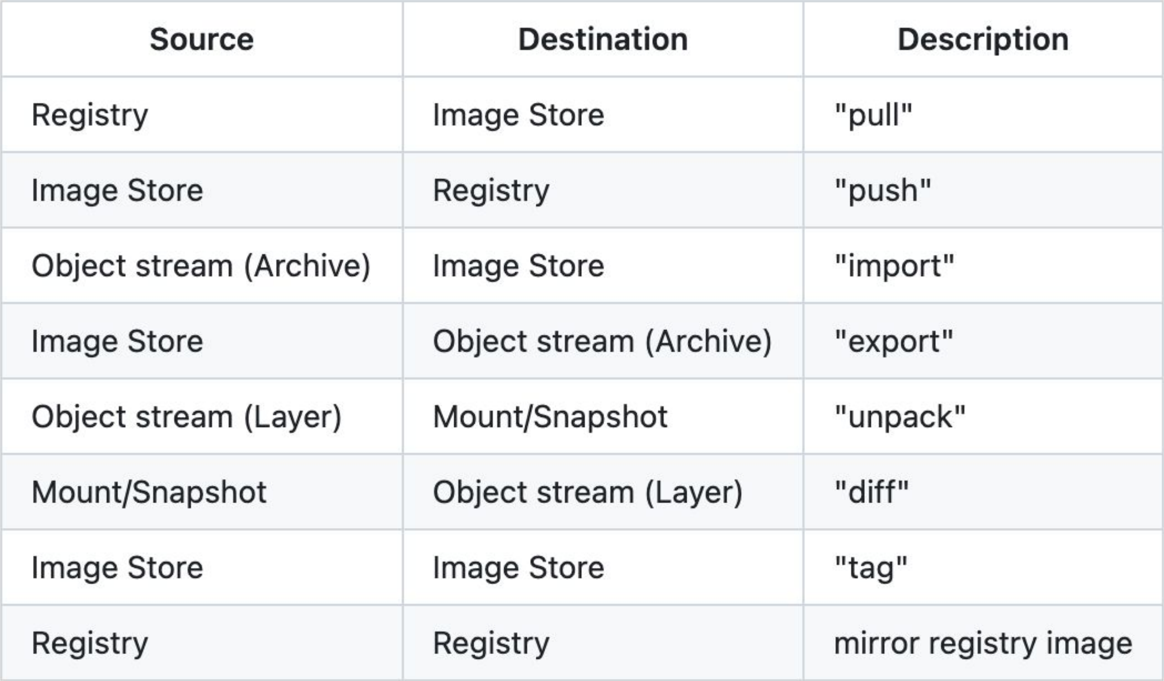


# Transfer Service



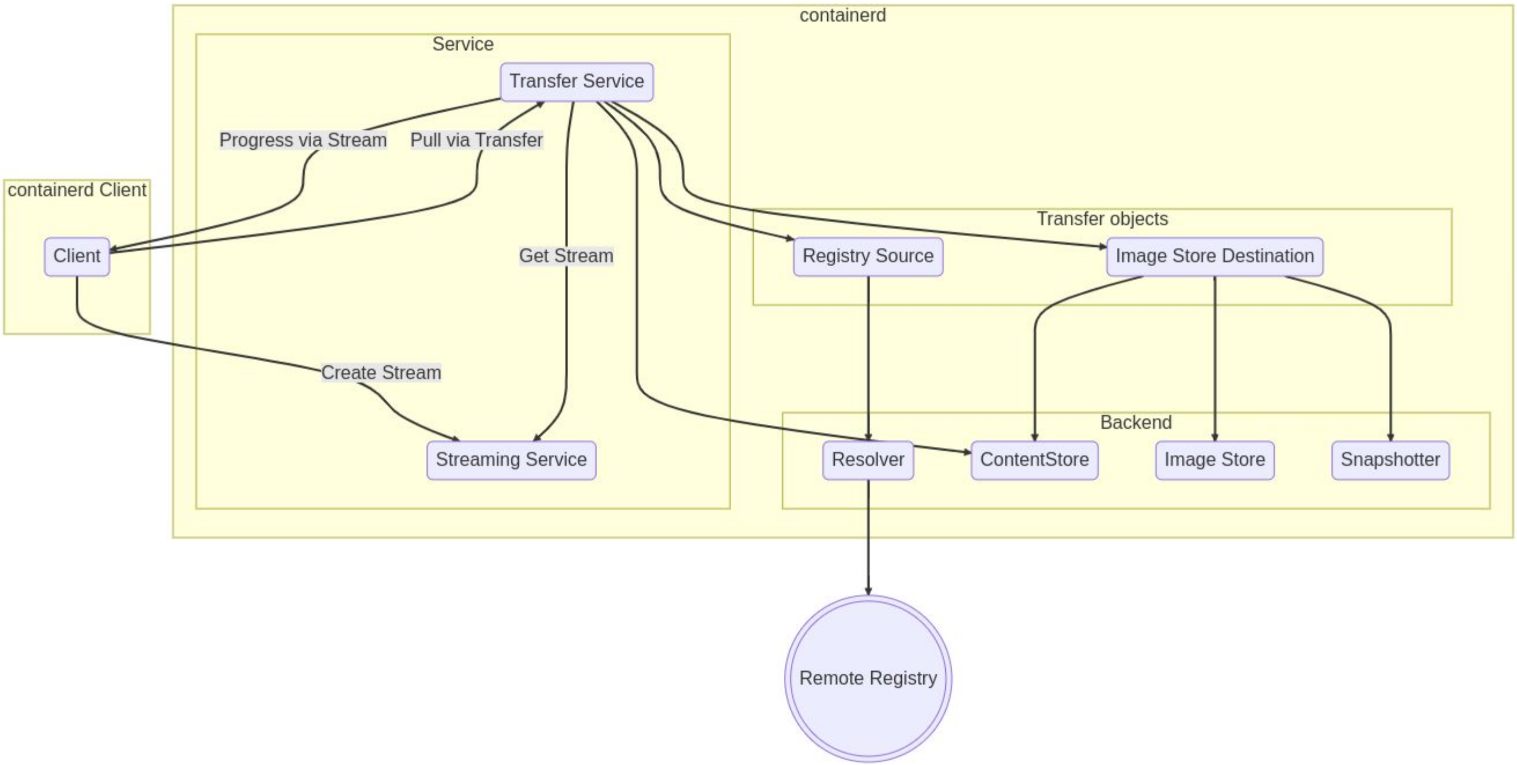
2019 - Kubecon San Diego

# Transfer Service



| Source | Destination | Description |
|---|---|---|
| Registry | Image Store | "pull" |
| Image Store | Registry | "push" |
| Object stream (Archive) | Image Store | "import" |
| Image Store | Object stream (Archive) | "export" |
| Object stream (Layer) | Mount/Snapshot | "unpack" |
| Mount/Snapshot | Object stream (Layer) | "diff" |
| Image Store | Image Store | "tag" |
| Registry | Registry | mirror registry image |

# Transfer Service



# Transfer Service

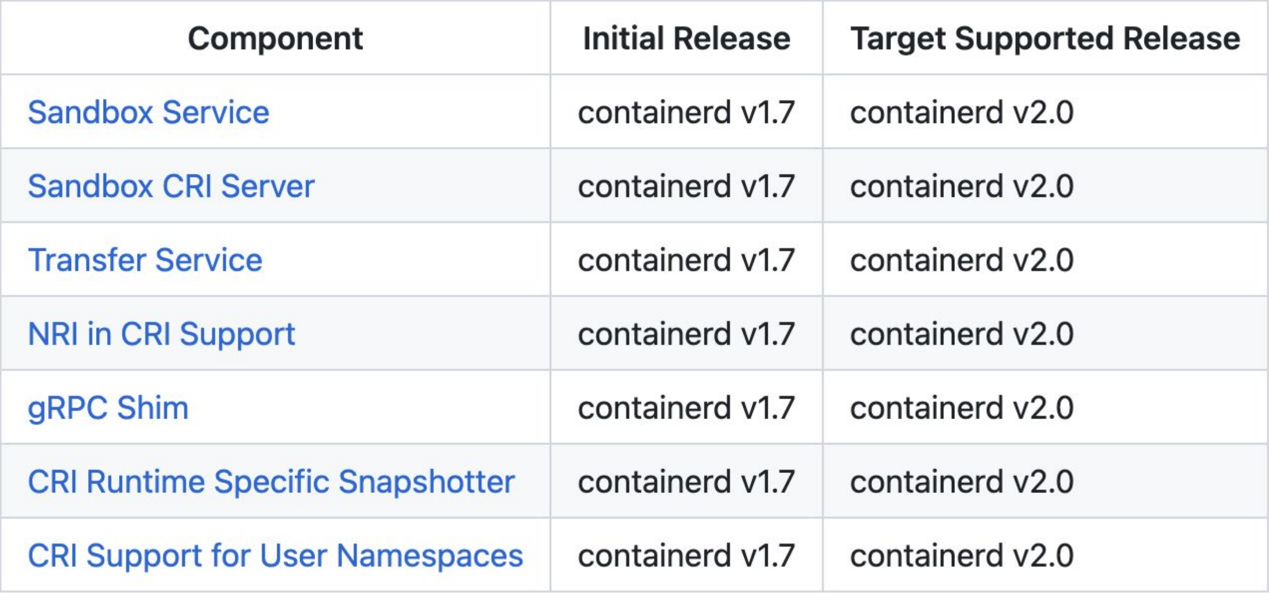
- New use-cases and extensions
  - OCI Referrers API support (mountable images, lazy-loading images)
  - Signing and image validation
    - [transfer] plugin to transfer service for image verification
    - Support Ratify as a containerd plugin
  - Confidential computing (guest sandbox env is the destination)
  - Customize image pulling logic
- Enable Transfer Service in CRI plugin by default

# User-Namespace Support

- **Support for user namespaces in stateless pods (v1.7)**
  - Only support emptyDir, configmap, secret, downwardsAPI
  - Use chown and cache the snapshots with same mapping
- **Supports Running Stateful Pods in (v2.0)**
  - Integrated with Idmapped mount (Merged in main branch!!!)
  - User Namespaces: Now Supports Running Stateful Pods in Alpha!

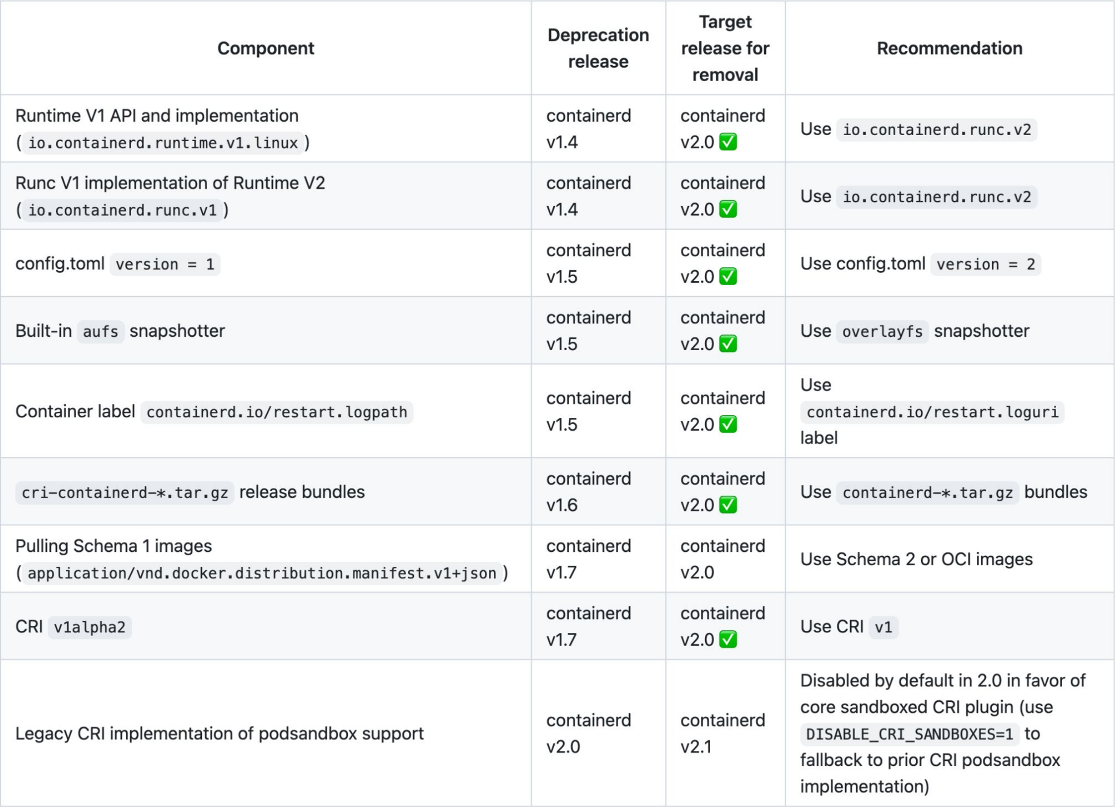
# v2.0 - Release Plan

- **Alpha/Beta**: 2023.11 (KubeCon + CloudNativeCon North America)
- **Beta**: 2023.12
- **GA**: 2024.2.10



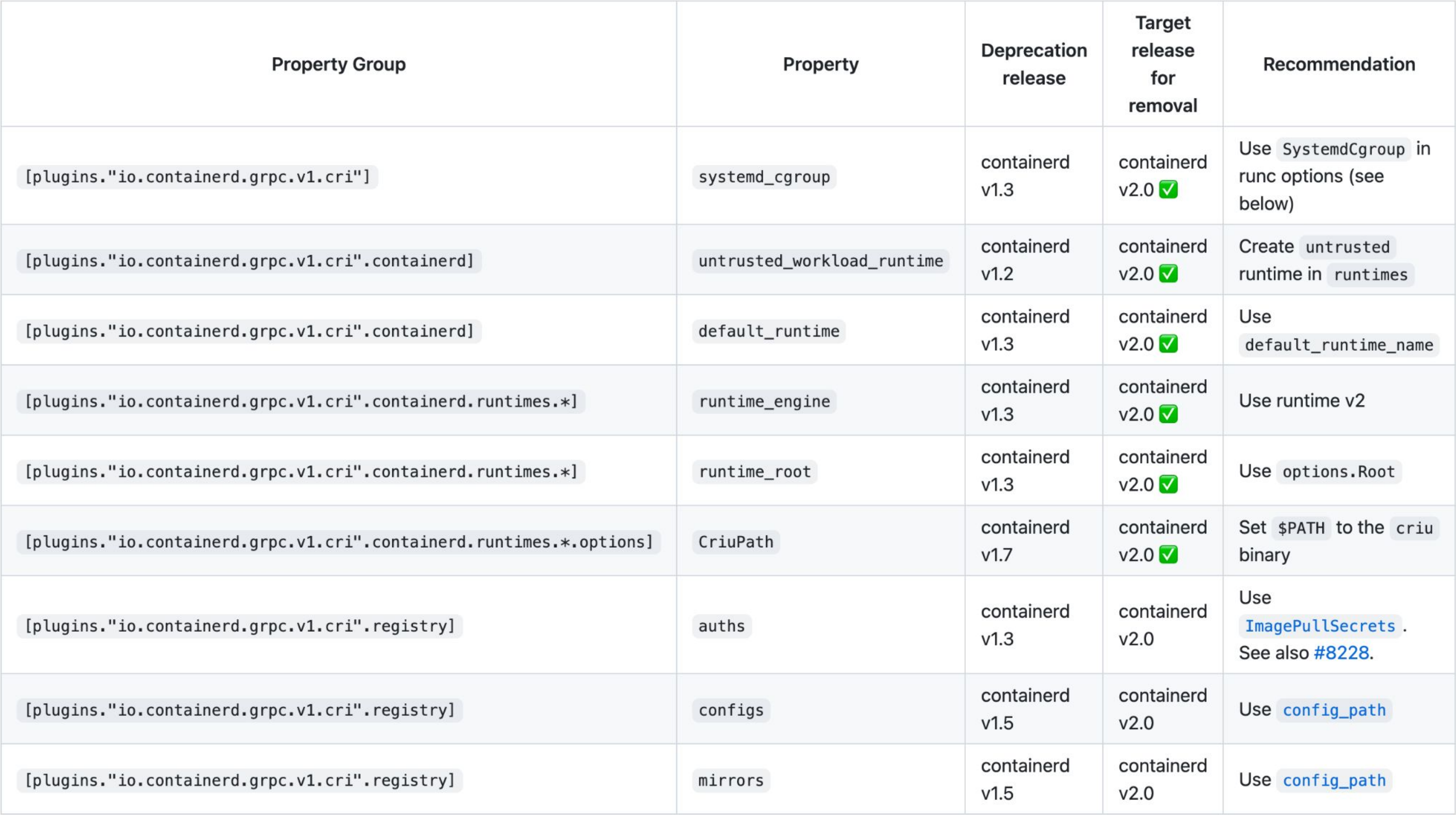
| Component | Initial Release | Target Supported Release |
|---|---|---|
| Sandbox Service | containerd v1.7 | containerd v2.0 |
| Sandbox CRI Server | containerd v1.7 | containerd v2.0 |
| Transfer Service | containerd v1.7 | containerd v2.0 |
| NRI in CRI Support | containerd v1.7 | containerd v2.0 |
| gRPC Shim | containerd v1.7 | containerd v2.0 |
| CRI Runtime Specific Snapshotter | containerd v1.7 | containerd v2.0 |
| CRI Support for User Namespaces | containerd v1.7 | containerd v2.0 |

# v2.0 - Removed Features



| Component | Deprecation release | Target release for removal | Recommendation |
|---|---|---|---|
| Runtime V1 API and implementation (`io.containerd.runtime.v1.linux`) | containerd v1.4 | containerd v2.0 ✅ | Use `io.containerd.runc.v2` |
| Runc V1 implementation of Runtime V2 (`io.containerd.runc.v1`) | containerd v1.4 | containerd v2.0 ✅ | Use `io.containerd.runc.v2` |
| config.toml `version = 1` | containerd v1.5 | containerd v2.0 ✅ | Use config.toml `version = 2` |
| Built-in `aufs` snapshotter | containerd v1.5 | containerd v2.0 ✅ | Use `overlayfs` snapshotter |
| Container label `containerd.io/restart.logpath` | containerd v1.5 | containerd v2.0 ✅ | Use `containerd.io/restart.loguri` label |
| `cri-containerd-*.tar.gz` release bundles | containerd v1.6 | containerd v2.0 ✅ | Use `containerd-*.tar.gz` bundles |
| Pulling Schema 1 images (`application/vnd.docker.distribution.manifest.v1+json`) | containerd v1.7 | containerd v2.0 | Use Schema 2 or OCI images |
| CRI `v1alpha2` | containerd v1.7 | containerd v2.0 ✅ | Use CRI `v1` |
| Legacy CRI implementation of podsandbox support | containerd v2.0 | containerd v2.1 | Disabled by default in 2.0 in favor of core sandboxed CRI plugin (use `DISABLE_CRI_SANDBOXES=1` to fallback to prior CRI podsandbox implementation) |

# v2.0 - Removed CRI Config Properties



| Property Group | Property | Deprecation release | Target release for removal | Recommendation |
|---|---|---|---|---|
| `[plugins."io.containerd.grpc.v1.cri"]` | `systemd_cgroup` | containerd v1.3 | containerd v2.0 ✅ | Use `SystemdCgroup` in runc options (see below) |
| `[plugins."io.containerd.grpc.v1.cri".containerd]` | `untrusted_workload_runtime` | containerd v1.2 | containerd v2.0 ✅ | Create `untrusted` runtime in `runtimes` |
| `[plugins."io.containerd.grpc.v1.cri".containerd]` | `default_runtime` | containerd v1.3 | containerd v2.0 ✅ | Use `default_runtime_name` |
| `[plugins."io.containerd.grpc.v1.cri".containerd.runtimes.*]` | `runtime_engine` | containerd v1.3 | containerd v2.0 ✅ | Use runtime v2 |
| `[plugins."io.containerd.grpc.v1.cri".containerd.runtimes.*]` | `runtime_root` | containerd v1.3 | containerd v2.0 ✅ | Use `options.Root` |
| `[plugins."io.containerd.grpc.v1.cri".containerd.runtimes.*.options]` | `CriuPath` | containerd v1.7 | containerd v2.0 ✅ | Set `$PATH` to the `criu` binary |
| `[plugins."io.containerd.grpc.v1.cri".registry]` | `auths` | containerd v1.3 | containerd v2.0 | Use `ImagePullSecrets`. See also #8228. |
| `[plugins."io.containerd.grpc.v1.cri".registry]` | `configs` | containerd v1.5 | containerd v2.0 | Use `config_path` |
| `[plugins."io.containerd.grpc.v1.cri".registry]` | `mirrors` | containerd v1.5 | containerd v2.0 | Use `config_path` |

# Expanded Ecosystem

- Built to be extensible
- Lots of places to plug in new functionality!
  - snapshotters
  - oci runtimes
  - runtime shims
  - clients
  - nri plugins
- New non-core projects are part of containerd
- A lot of adaptions from community project, vendor products.

# Kubernetes distros adopting containerd

- Alibaba Cloud Container Service for Kubernetes
- Amazon Elastic Kubernetes Service
- Azure Kubernetes Service
- Google Kubernetes Engine
- Huawei Cloud Cloud Container Engine
- IBM Cloud Kubernetes Service
- Rancher K3s
- VMware Tanzu
- Volcengine Kubernetes Engine

# Containerd Clients

- **ctr -** command-line development tool, core containerd project
- **nerdctl** - non-core containerd project - a Docker-like CLI
  - expanded functionality  eg. Lazy-loading images, image encryption, image signing
- **crictl** - a CLI for CRI - Kubernetes project (part of cri-tools)

- **Colima** - container runtimes on macOS (and Linux) with minimal setup
- **Finch** - Docker-like CLI on MacOS
- **Rancher Desktop** - Docker-like experience on MacOS, Windows, and Linux

# Snapshotters

- **Builtin**
  - overlayfs (Linux)
  - devmapper (Linux)
  - btrfs (Linux)
  - native (Linux/Unix/Windows)
  - blockfile (New! Linux/Unix)
  - zfs (Linux/Unix)
  - LCOW (Windows)
  - Windows (Windows)

- **Extension via proxy plugins**

- **Remote - Lazy Loading**
  - stargz (Filesystem, non-core project)
  - overlaybd (Block, non-core project)
  - nydus (Filesystem, non-core project)

  - SOCI (Filesystem, OSS vendor project)
  - GKE image streaming (Filesystem, vendor project)

# Runtimes & Shims

- **runc** - standard OCI runtime for Linux containers
- **crun** - alternative OCI runtime for Linux containers, written in C
- **youki** - alternative OCI runtime for Linux containers, written in Rust
- **runj** - experimental OCI runtime for FreeBSD jails

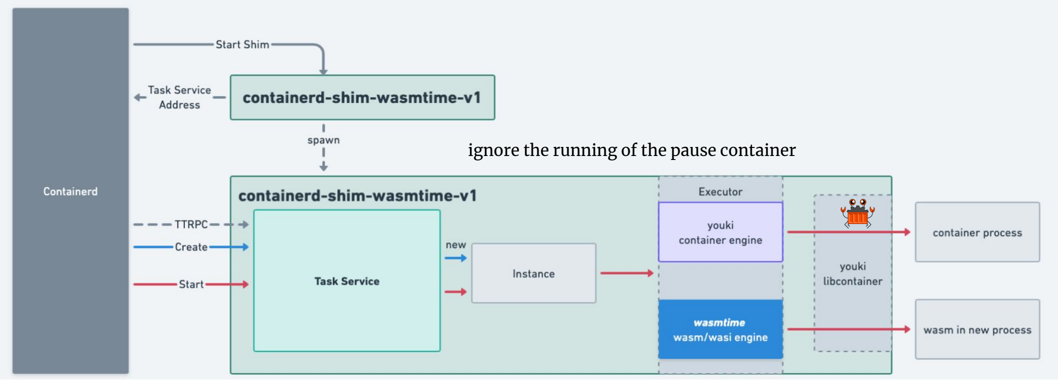
- **hcsshim/runhcs** - containerd shim and OCI runtime for Windows containers
- **runwasi** - (New! Non-core project) - OCI runtime for WASM
- **Kata Containers** - hypervisor-based isolation for pods
- **gVisor/runsc** - independent kernel for isolation
- **firecracker-containerd** - hypervisor-based isolation for containers based on Firecracker
- **inclavare-containers** - run containers in hardware-assisted Trusted Execution Environment (TEE)
- **kuasar** - an container runtime supporting multiple sandbox techniques.

- **embedshim** - An ebpf-based container task runtime manager

# Containerd & WASM: runwasi

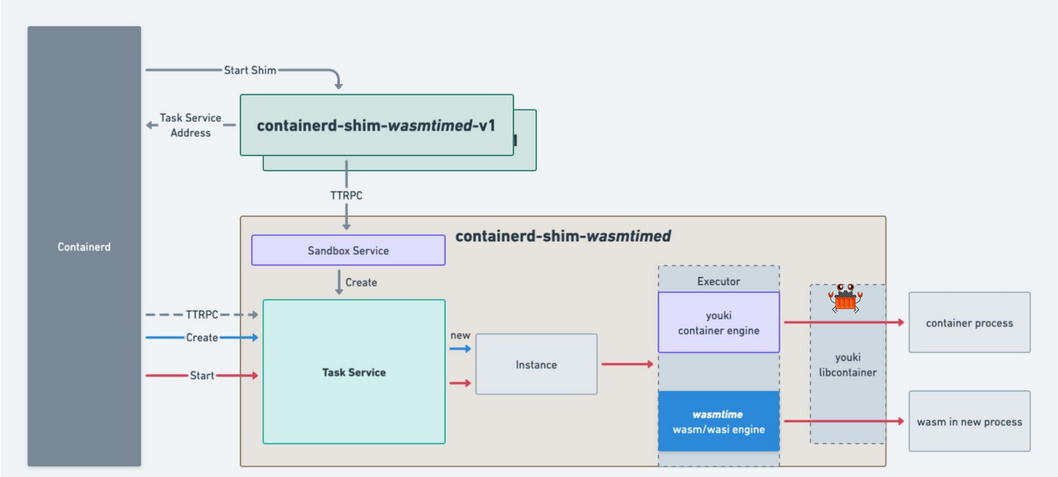
https://github.com/containerd/runwasi/tree/ab2158ffce953b51c996e516bc61e3eaa39ba3c1

## Normal Mode



ignore the running of the pause container

## Shared Mode



⚠️ Alpha quality software, do not use in production.

# Containerd & WASM: wasm-shims

**containerd/runwasi:**

- containerd-shim-wasmtime-v1
- containerd-shim-wasmedge-v1
- containerd-shim-wasmer-v1

**deislabs/containerd-wasm-shims:**

- containerd-shim-spin-v1
- containerd-shim-slight-v1
- containerd-shim-lunatic-v1
- containerd-shim-wws-v1

**Kwasm**: Install WASM support on your Kubernetes Nodes ⚠️Only for development or evaluation purpose

```
kubectl annotate node kind-worker2 kwasm.sh/kwasm-node=true
```

wasm shims:

- containerd/runwasi:
  - *containerd-shim-[wasmtime,wasmedge,wasmer]-v1*

- deislabs/containerd-wasm-shims:
  - *containerd-shim-[spin,slight,lunatic,wws]-v1*

# Containerd & WASM: WG-WASM

**WASM OCI Artifacts**

- https://docs.google.com/document/d/11shgC3l6gplBjWF1VJCWvN_9do51otscAm0hBDGSSAc/edit
- https://github.com/containerd/containerd/pull/8699

## Proposal：Containerd shim lifecycle operator & Shim CRD

- https://hackmd.io/TwC8Fc8wTCKdoWlgNOqTgA

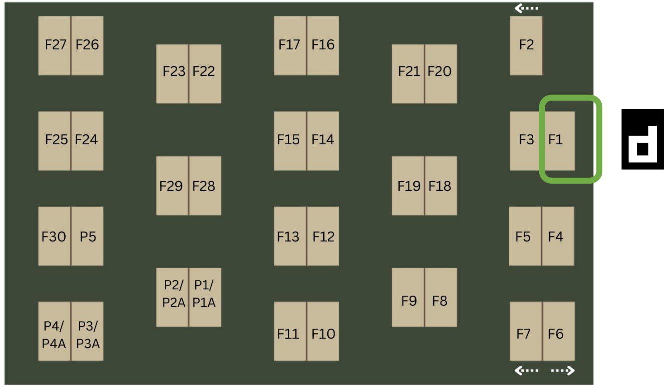
# Getting involved

- #containerd and #containerd-dev channel on
  - CNCF Slack (https://slack.cncf.io)
- **Community Meeting on the second Thursday each month**
  - See CNCF Calendar for your timezone (https://cncf.io/calendar)
- Build something in the ecosystem!
- Discussion, issues and pull requests welcome!
  - https://github.com/containerd/containerd

# Thank you

KubeCon | CloudNativeCon
OPEN SOURCE SUMMIT
China 2023

PROJECT PAVILION FLOORPLAN



**Full Time Kiosk**

containerd F1
Prometheus F2
Kube-ovn F20
KubeArmor F21
Merbridge F22
open-cluster-management F23
ORAS F24
PipeCD F25
Pravega F26
SlimToolkit F27
Piraeus Datastore F28
Vineyard F29
Istio F3
WasmEdge F30
CubeFS F5

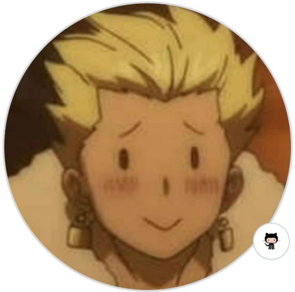
**Full Time Kiosk**

TiKV F4
KubeEdge F6
Kyverno F7
Longhorn F8
Notary F9
OpenKruise F10
Volcano F11
Aeraki Mesh F12
Antrea F13
Carina F14
Clusterpedia F15
FebEdge F16
hwameistor F17
K3s F18
Karmada F19

**Part-Time Kiosk**

Kubernetes P1
Harbor P1A
kubespray P2
SIG Node P2A
Cilium P3
Chaos Mesh P3A
Porter P4
Kepler P4A
Paralus P5

**Fu Wei**
fuweid · he/him



**Iceber Gu**
Iceber

**Wednesday, 27 September: 10:30 - 13:30**
**Wednesday, 27 September: 14:30 - 18:45**

**Thursday, 28 September: 10:30 - 14:00**