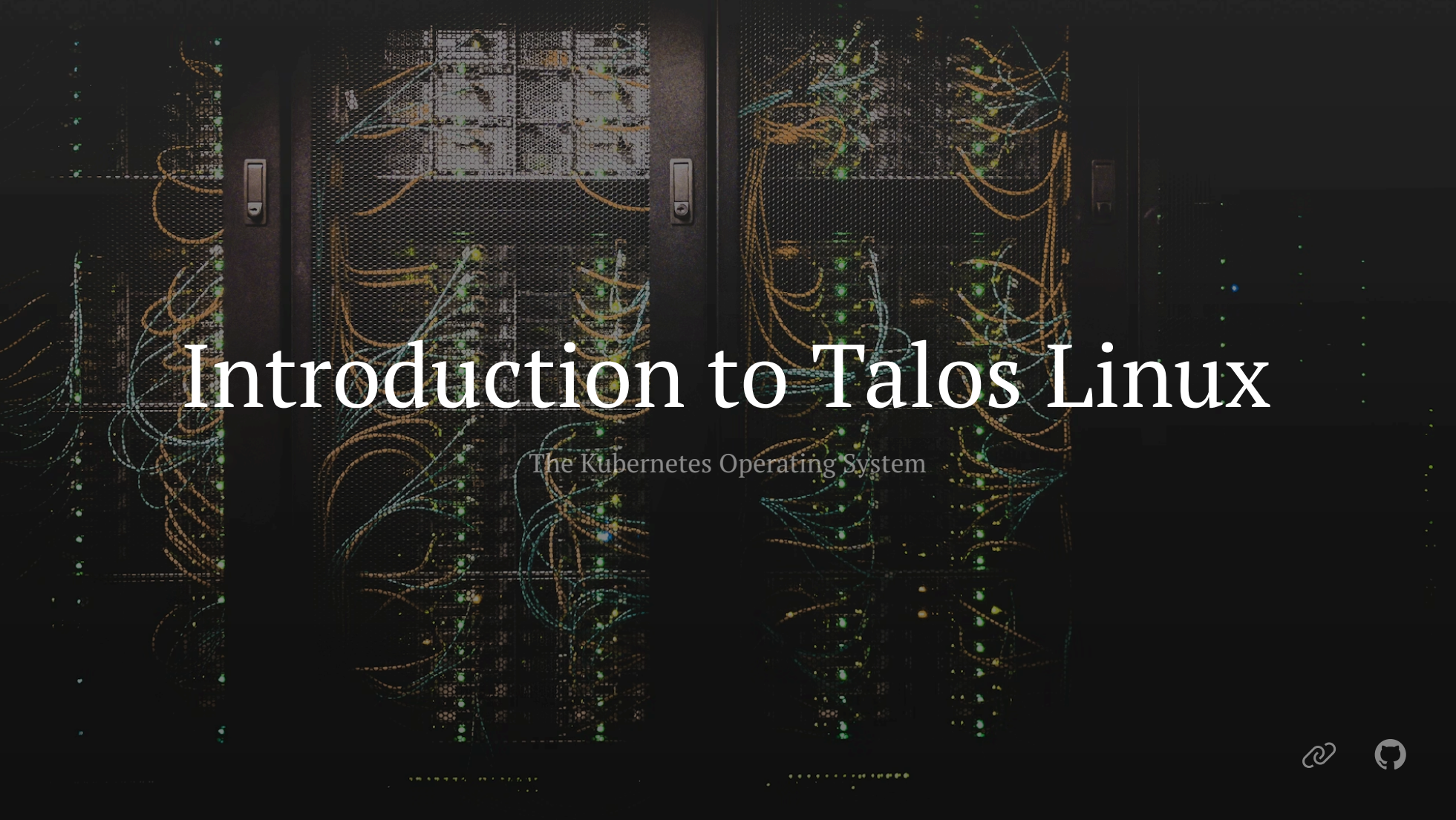
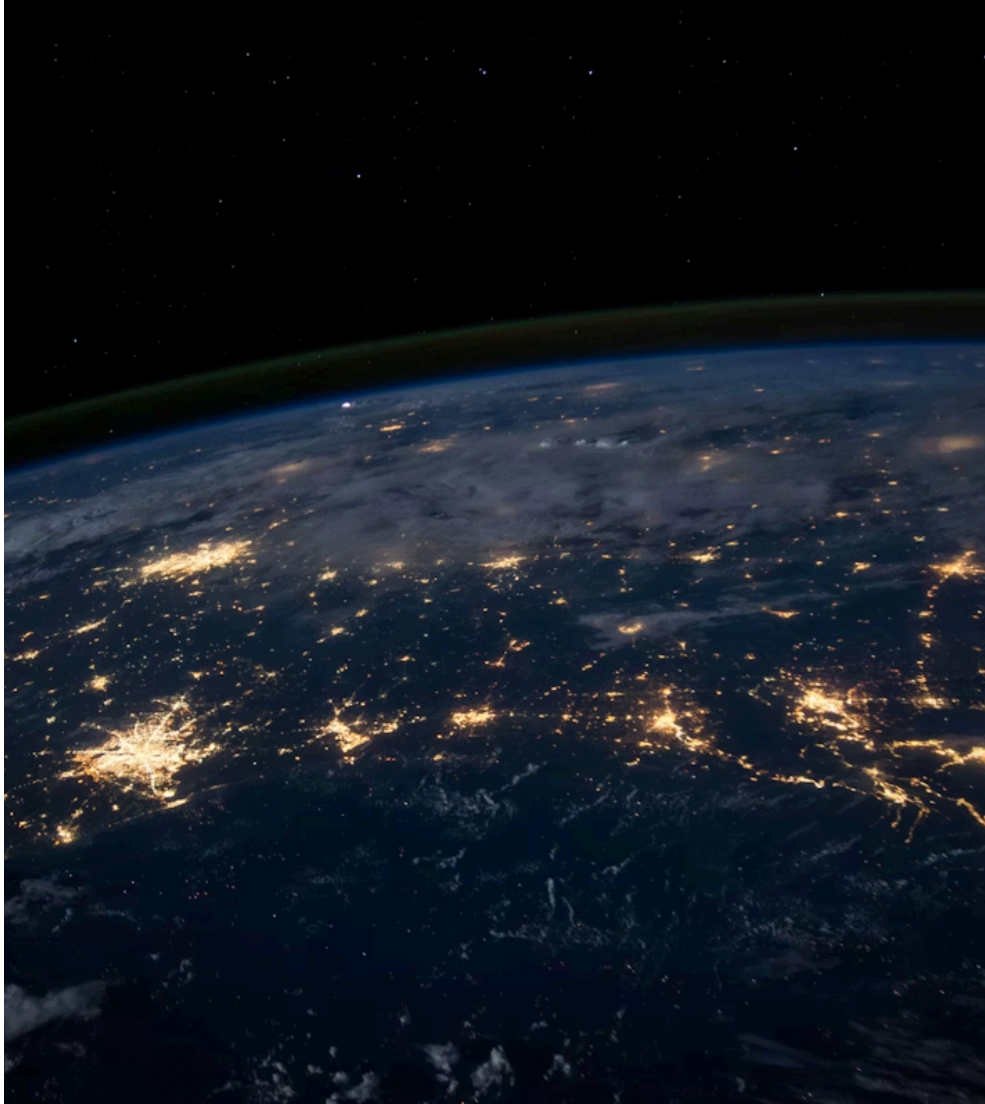


Introduction to Talos Linux

The Kubernetes Operating System

What is Talos Linux?

- A **modern Linux distribution** built specifically for Kubernetes
- Developed by **Sidero Labs**
- **Not based on any other distribution** - built from scratch
- Runs the Linux kernel with custom userspace written in **Go**
- **No shell, no SSH, no packages** - everything is API-driven
- Designed to be **secure, immutable, and minimal**



Why Talos Linux?

Traditional Linux distributions come with:

- SSH access (attack surface)
- Package managers (dependency hell)
- Configuration drift
- Manual maintenance
- Security vulnerabilities
- Shell access for attackers

Talos Provides

- **No SSH** - API-only access
- **No packages** - single immutable image
- **No drift** - declarative configuration
- **Automatic updates** - atomic and reversible
- **Minimal attack surface**
- **No shell** for attackers to exploit

Core Design Principles

Secure

Minimal attack surface with mTLS authentication

Immutable

Read-only root filesystem prevents tampering

Minimal

Only essential components for Kubernetes

Key Features

API-Driven Management

All operations through secure gRPC API

Immutable Infrastructure

Read-only root filesystem

Declarative Configuration

YAML-based reproducible deployments

Automatic Updates

Atomic updates with rollback

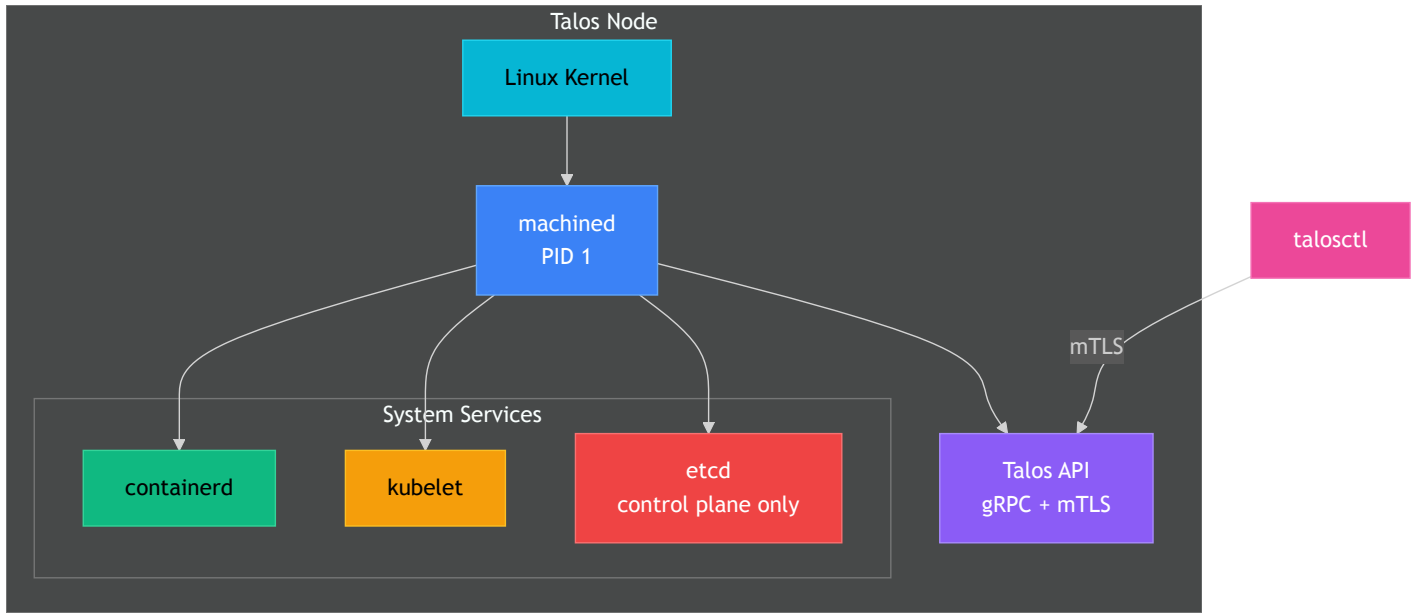
Multi-Platform Support

Bare metal, cloud, virtualization

Built for Kubernetes

Optimized from the ground up

Architecture Overview



No systemd, no shell, no SSH - just what's needed for Kubernetes

Security Model

No SSH Access

- Eliminates remote shell attacks
- All access via authenticated API

Mutual TLS (mTLS)

- All API communications encrypted
- Client certificate authentication

Signed Kernel Modules

- Cryptographically validated
- Tamper-proof kernel

Read-Only Root

- Prevents filesystem modifications
- Immutable system state

```yaml
# Example: Role-based API access
machine:
  features:
    kubernetesTalosAPIAccess:
      enabled: true
      allowedRoles:
        - os:reader
        - os:admin
      allowedKubernetesNamespaces:
        - kube-system
```

Security by design, not by configuration

Immutability Explained

Traditional Linux

- Mutable filesystem
- Package updates change system
- Configuration drift over time
- Difficult to reproduce state
- Security patches are additive

Talos Linux

- Read-only root filesystem
- Atomic image-based updates
- Consistent state always
- 100% reproducible
- Whole-image security updates

Managing Talos with talosctl

The `talosctl` CLI is your primary interface to Talos clusters.

```
# Get cluster information
talosctl get members

# Apply configuration
talosctl apply-config --nodes 10.0.0.1 -f config.yaml

# View services status
talosctl services

# Access logs
talosctl logs kubelet
```

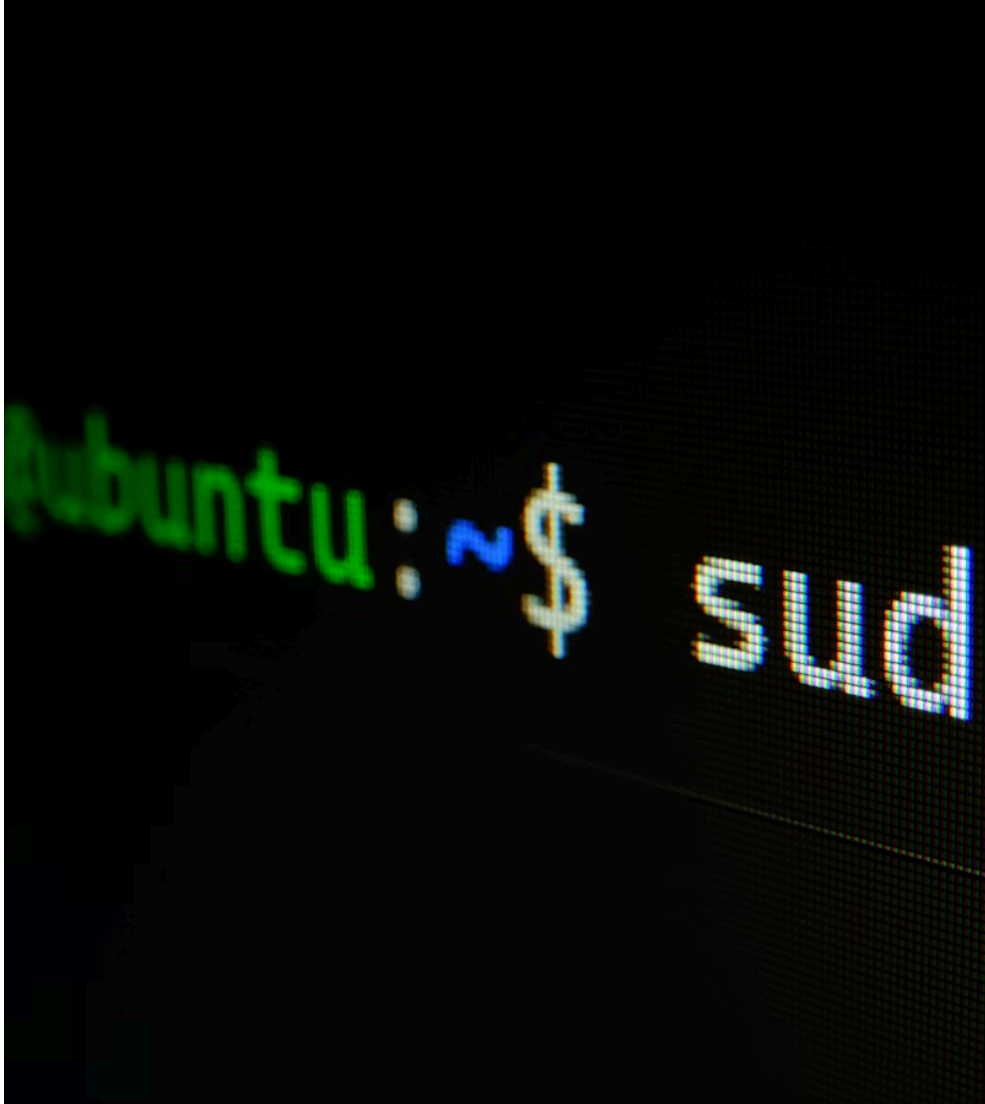
```
# Upgrade Talos
talosctl upgrade --nodes 10.0.0.1 \
  --image ghcr.io/siderolabs/installer:v1.11.0

# Get resources (kubectl-like)
talosctl get rd  # resource definitions

# Dashboard view
talosctl dashboard
```

ⓘ No need for SSH - everything is done through the API

Configuration Example

```yaml
version: v1alpha1
machine:
  type: controlplane
  certSANs:
    - 10.0.0.1
    - talos.local
  kubelet:
    image: ghcr.io/siderolabs/kubelet:v1.32.0
  network:
    hostname: talos-cp-1
    interfaces:
      - interface: eth0
        dhcp: true
cluster:
  clusterName: my-cluster
  controlPlane:
    endpoint: https://10.0.0.1:6443
  network:
    cni:
      name: cilium
```

→ Declarative, version-controlled, reproducible



Platform Support

Bare Metal

ISO, PXE boot, USB

Cloud Providers

AWS, GCP, Azure, DigitalOcean, Hetzner

Virtualization

VMware, Proxmox, QEMU, Hyper-V

SBCs

Raspberry Pi, Jetson Nano

Edge

IoT and edge computing

Use Cases

Production Kubernetes

Run secure, maintainable Kubernetes clusters with predictable behavior and easy upgrades.

Edge Computing

Lightweight and secure for resource-constrained edge environments.

Home Labs

Perfect for learning Kubernetes with a production-grade experience.

High-Performance Computing

Minimal overhead maximizes resources for compute-intensive workloads.

KubeSpan

Built-in WireGuard mesh networking for secure cluster communication.

What is KubeSpan?

- **Encrypted mesh network** using WireGuard
- **Automatic peer discovery** via cluster discovery service
- **Zero configuration** - just enable it

Benefits

- Secure node-to-node communication
- Works across clouds and on-premises
- NAT traversal built-in
- No external VPN required

```yaml
# Enable KubeSpan
machine:
  network:
    kubespan:
      enabled: true
cluster:
  discovery:
    enabled: true
```

```
# Check KubeSpan status
talosctl get kubespanpeerspecs
talosctl get kubespanpeerstatuses
```

Secure multi-cloud clusters made easy

Getting Started

1. Install talosctl

```
# macOS
brew install siderolabs/tap/talosctl

# Linux
curl -sL https://talos.dev/install | sh
```

2. Generate Configuration

```
talosctl gen config my-cluster \
  https://10.0.0.1:6443
```

3. Apply Configuration

```
talosctl apply-config \
  --insecure \
  --nodes 10.0.0.1 \
  --file controlplane.yaml
```

4. Bootstrap Cluster

```
talosctl bootstrap \
  --nodes 10.0.0.1

# Get kubeconfig
talosctl kubeconfig
```

Your secure Kubernetes cluster is ready in minutes!

Talos vs Traditional Linux

Feature	Traditional Linux	Talos Linux
Shell Access	Yes (bash, sh)	None
SSH	Enabled by default	Not available
Package Manager	apt, yum, etc.	None - single image
Init System	systemd	machined (custom)
Configuration	Files, scripts	Declarative YAML
Updates	Package-by-package	Atomic image updates
Root Filesystem	Read-write	Read-only

Key Takeaways

Security First

No SSH, mTLS everywhere, signed kernels, minimal attack surface

Predictable

Immutable, declarative, reproducible deployments

Kubernetes Native

Built from scratch specifically for Kubernetes

Learn More

Documentation

talos.dev

GitHub

siderolabs/talos

Quick Start

Get started guide

Developed by **Sidero Labs**

Thank You!

Questions?

Visit talos.dev

GitHub Repo