

CNCF-Grundlagen

**Was die CNCF ist**

2015 als Subsidiary der Linux Foundation gegründet, Seed-Projekt Kubernetes (von Google gestiftet). Heutige Aufgabe: neutrale Heimat für Cloud-Native-Projekte, mit Governance, Marketing und Events.

TOC (Technical Oversight Committee): definiert Reifegrade, führt Due-Diligence, stimmt über Level-Wechsel ab.

TAGs (Technical Advisory Groups: Security, Runtime, Observability, App Delivery) lösten die früheren SIGs ab; die Cloud Native AI WG sitzt unter TAG Runtime.

KubeCon + CloudNativeCon: Flaggschiff-Konferenz.

**Kern-Zahlen (Stand Mitte 2026)**

37 Graduated + 38 Incubating + 150 Sandbox = 225 aktive CNCF-gehostete Projekte, dazu 28 Archived (read-only).

Ökosystem über die erste Dekade: >300.000 Contributors, >11.500 Organisationen, 190 Länder, 240+ Projekte im Landscape.

Die Zahlen sind ein bewegliches Ziel, vor Zitat an der Quelle gegenprüfen.

**# An der Quelle pruefen, nicht auswendig:**

|                                |                                       |
|--------------------------------|---------------------------------------|
| <b>landscape.cncf.io</b>       | # Landscape, Filter nach Maturity     |
| <b>cncf.io/project-metrics</b> | # Live-Zahlen, Velocity, Contributors |

**contribute.cncf.io/projects/lifecycle** # Level-Kriterien

**github.com/cncf/toc** # Due-Diligence, Level-Votes

**Reifegrade (Maturity Ladder)**

**Sandbox, Incubating, Graduated, Archived**

Sandbox: experimenteller Einstieg, keine Due-Diligence. Verlangt Code-of-Conduct, IP-/Governance-Policy und Adoptions-Potenzial.

Incubating: ≥3 unabhängige Produktions-Adopter (**ADOPTERS.md**), gesunde Change-Rate, Committer aus mehreren Orgs, TOC-Due-Diligence.

Graduated: zusätzlich Committer aus ≥2 Orgs, OpenSSF Best Practices Badge (passing), abgeschlossenes unabhängiges Third-Party-Security-Audit, explizite Governance, TOC-Supermajority (2/3).

Archived: inaktiv, read-only.

**Was der Reifegrad nicht sagt**

Das Level misst Projekt-Governance und Adoptionsbreite, nicht Feature-Umfang, Qualität oder Eignung für deinen Stack. Sandbox kann längst produktiv laufen (k3s), ein Graduated-Projekt kann für deinen Kontext falsch sein. Level als Risiko-Signal lesen, nicht als Kaufempfehlung.

**Graduated-Projekte (37)**

**Orchestrierung, Runtime, Control-Plane**

Kubernetes Orchestrierung (Seed) · etcd KV-Store der Control-Plane · CoreDNS DNS/Service-Discovery · containerd + CRI-O Container-Runtimes (CRI) · KEDA event-getriebenes Autoscaling · Knative Serverless/Eventing (2025) · Crossplane Control-Plane/IaC via CRDs (2025) · KubeEdge Kubernetes für Edge/IoT.

**Netzwerk & Service Mesh**

Envoy L4/L7-Proxy, Mesh-Datenebene · Cilium eBPF für Networking, Security, Observability (CNI) · Istio Service Mesh (Envoy-basiert) · Linkerd leichtes Mesh (Rust-Proxy).

**Observability**

Prometheus Metriken & Alerting (PromQL) · OpenTelemetry Telemetrie-Standard für Traces/Metrics/Logs (graduated 2026-05, Velocity #2 nach Kubernetes) · Jaeger Distributed Tracing · Fluentd Log-Collection & -Routing.

**Delivery, Packaging & Registry**

Argo GitOps-CD, Workflows, Rollouts · Flux GitOps-CD (controller-nativ) · Helm Paketmanager (Charts) · Buildpacks Quellcode zu OCI ohne Dockerfile<sup>†</sup> · Harbor OCI-Registry mit Scan/Signing · Dragonfly P2P-Image-Distribution.

**Security & Supply-Chain**

OPA Policy-Engine (Rego, Gatekeeper) · Kyverno K8s-native Policy (2026-03) · Falco Runtime-Security/Threat-Detection · cert-manager X.509-Automation · SPIFFE + SPIRE Workload-Identity (SVID) · TUF sicheres Update-Framework · in-toto Supply-Chain-Attestierung.

**Daten, Storage & App-Runtime**

Vitess horizontal skalierendes MySQL (Sharding) · TiKV transaktionaler KV-Store · Rook Storage-Orchestrator (Ceph auf K8s) · CephFS verteiltes Cloud-Native-Filesystem · Dapr Distributed-Application-Runtime · CloudEvents Standard für Event-Daten.

<sup>†</sup> Buildpacks: Graduierung Mitte 2026, jüngster Zugang, Stand mit Vorbehalt.

Landscape-Taxonomie

G=Graduated, I=Incubating, S=Sandbox; ohne Kürzel = kein CNCF-Projekt (Quelle in Klammern)

**Provisioning & Runtime**

Automation: KubeEdge (G), OpenTofu (S), Cluster API (K8s-SIG), Ansible.

Registry: Harbor (G), Dragonfly (G).

Security: OPA (G), Kyverno (G), Falco (G), in-toto (G), TUF (G), Notary Project (I), Kubescape (I), Sigstore (OpenSSF), Trivy (Aqua).

Key Mgmt: SPIFFE/SPIRE (G), External Secrets Operator (S), Vault (HashiCorp).

Storage: Rook (G), CephFS (G), Longhorn (I).

Runtime/Network: containerd (G), CRI-O (G), Cilium (G), gVisor/Kata, Calico.

**Orchestration & Management**

Scheduling: Kubernetes (G), KEDA (G), Knative (G), Crossplane (G), Karmada (I), Volcano (I), Kubeflow (I).

Coordination: etcd (G), CoreDNS (G).

Proxy/Gateway: Envoy (G), Contour (I), Emissary-Ingress (I), Gateway API (K8s-SIG).

Service Mesh: Istio (G), Linkerd (G), Kuma (S).

**App Definition & Development**

Database: Vitess (G), TiKV (G).

Streaming: CloudEvents (G), NATS (I), Strimzi (I).

App-Def & Build: Helm (G), Dapr (G), Buildpacks (G), Backstage (I), KubeVela (I), Operator Framework (I), Score (S), Kustomize.

CI/CD: Argo (G), Flux (G), Tekton (I).

**Observability, Platform & KI**

Observability: Prometheus (G), OpenTelemetry (G), Jaeger (G), Fluentd (G), Thanos (I), Cortex (I).

Chaos/Cost: Chaos Mesh (I), Litmus (I), OpenCost (I), OpenFeature (I).

Platform: k3s (S), k0s (S), Flatcar (I).

Wasm: wasmCloud (I), SpinKube (S), WasmEdge (S).

KI (neu 2025/26): AI Agent, Inference (KServe I), Data, Training, AI Native Infra.

Auswahl-Achsen

**Runtime & CNI**

Runtime: containerd (Default) · CRI-O (OpenShift-nah) · gVisor/Kata für Sandbox-Isolation.

CNI: Cilium (De-facto-Default, eBPF, Hubble) · Calico (policy-stark) · Flannel (simpel, ohne NetworkPolicy).

**Service Mesh, Ingress & Gateway API**

Mesh: Istio (Vollausbau, Ambient/sidecarless) · Linkerd (leicht, Rust) · Cilium mesh (wenn CNI schon Cilium ist).

Ingress: Gateway API (K8s-SIG, GA) löst Ingress ab; klassisches NGINX-Ingress ist deprecated, EoL März 2026. Impl.: Envoy Gateway, Istio, Cilium, Kong, Contour. GAM-MA für Mesh-Ost-West.

**GitOps & Packaging**

GitOps: Argo CD (UI, Multi-Cluster, App-of-Apps) · Flux (leicht, controller-nativ, SOPS).

Packaging: Helm (Templating) · Kustomize (Overlays, kein Templating) · KubeVela/Score als Higher-Level-Abstraktion.

**Observability-Stack**

Metriken: Prometheus + OpenTelemetry (Instrumentierung/Collector).

Traces: Jaeger.

HA/Langzeit: Thanos oder Cortex.

UI/Logs: Grafana, Loki, Tempo, Mimir (Grafana Labs, kein CNCF-Level). Prometheus als Kern, der Rest ist austauschbar.

**Policy, Security & Secrets**

Policy: Kyverno (YAML-nativ, K8s-only) · OPA-Gatekeeper (Rego, plattformübergreifend).

Runtime: Falco · Tetragon (Cilium/eBPF, mit Enforcement).

Certs/Identity: cert-manager · SPIFFE/SPIRE.

Secrets: External Secrets Operator · Vault (HashiCorp).

Scan: Trivy (Aqua), Signing über Sigstore (OpenSSF).

**Serverless, Autoscaling & Provisioning**

Serverless/Eventing: Knative · KEDA (Event-Autoscaling) · CloudEvents.

Autoscaling: HPA/NPA (Pod) · Cluster-Autoscaler oder Karpenter (Node, just-in-time) · KEDA (event).

Cluster-Provisioning: Cluster API (K8s-SIG, deklarativ) · kubeadm/kOps · Karmada für Multi-Cluster.

**Developer-Platforms (IDP) & Registry**

IDP: Backstage (Developer-Portal, I) · Crossplane (Control-Plane/IaC, G) · KubeVela/Score als Deployment-Abstraktion.

Registry: Harbor (Enterprise, Scan/Signing) · Dragonfly (P2P-Distribution für große Images).

Häufige Fehlannahmen

**Was oft falsch zugeordnet wird**

Sigstore ist kein CNCF-Projekt, sondern OpenSSF (ebenfalls Linux Foundation). Häufigster Irrtum.

Cluster API ist ein K8s-SIG-Cluster-Lifecycle-Subprojekt, kein eigenes CNCF-Projekt.

Backstage ist Incubating, nicht Graduated.

Notary Project ist Incubating.

OpenTelemetry ist seit 2026-05 Graduated (früher oft noch als Incubating zitiert).

Karpenter, Trivy, Kueue, LeaderWorkerSet stehen im Landscape, haben aber kein CNCF-Level (Vendor-OSS bzw. K8s-SIG).

Trends 2024–2026

Real, nicht Hype

Platform Engineering/IDP: breite Absicht, aber Reifelücke, interne Backstage-Adoption bleibt oft niedrig.  
eBPF ist Mainstream: Cilium als Default bei GKE Autopilot, Tetragon liefert Detection und Enforcement im Kernel (Kernel 6.1+).  
Gateway API löst Ingress ab (v1.4 seit 2025-11), NGINX-Ingress EoL März 2026, GAM-MA (Mesh) ist GA.

Supply-Chain & Compliance

Getrieben von USEO 14028 und EU CRA: SBOM wird Baseline, SLSA v1.2 als Reifegrad-Modell, Signing über Sigstore (OpenSSF). CNCF-Seite: in-toto (G), TUF (G), Notary Project (I). Kein Kür mehr, sondern regulatorische Pflicht.

AI/ML & WASM auf Kubernetes

Cloud Native AI wächst am schnellsten: KServe (I) für Serving, Kueue als Batch-Standard, LeaderWorkerSet für Multi-Host-Inference, DRA (Dynamic Resource Allocation) GA in K8s 1.34 für topologie-bewusstes GPU-Scheduling, neue KI-Landscape-Kategorien.  
WASM bleibt nischig (SpinKube S, wasmCloud I): Edge/FaaS, Blocker sind fehlendes WASI-Threading und schwache Observability. Kein Container-Ersatz, eher daneben.



cloudnative-cheatsheet.de

Cloud Native Landscape von OMNI52

Aus 225 Projekten den passenden Stack schneiden

#### Architektur-Reviews & Tool-Auswahl

OMNI52™ hilft Plattform-Teams, aus der CNCF-Landschaft die passenden Bausteine zu wählen und ohne Produktions-Riss zu betreiben.

Runtime, CNI, Service Mesh, GitOps, Observability-Stack, Policy und Supply-Chain, bewertet auf eurer Topologie und eurem Compliance-Rahmen. Output landet als GitOps-Repo bei euch im Cluster: Helm-Charts, Runbooks, Diagnose-Skripte. Euer Team operiert weiter, nicht wir.

OMNI52 GmbH, Ebern · [istio-consulting.de](https://istio-consulting.de) · kostenloses 30-min Initial-Assessment

#### Verwandte Cheatsheets

Ebenfalls von OMNI52: [kubernetes-cheatsheet.de](https://kubernetes-cheatsheet.de) (Plattform-Layer) · [service-mesh-cheatsheet.de](https://service-mesh-cheatsheet.de) (Istio, Linkerd, Cilium im Vergleich) · [cilium-cheatsheet.de](https://cilium-cheatsheet.de) (eBPF/CNI) · [prometheus-cheatsheet.de](https://prometheus-cheatsheet.de) (Observability) · [argocd-cheatsheet.de](https://argocd-cheatsheet.de) & [flux-cheatsheet.de](https://flux-cheatsheet.de) (GitOps).

#### Lizenz & Weiterverteilung

CC BY-SA 4.0: du darfst dieses Cheatsheet kopieren, weiterverteilen, ausdrucken und in eigenen Materialien zitieren. Bedingung: Quellenangabe "Cloud Native Landscape Cheatsheet, OMNI52 GmbH, cloudnative-cheatsheet.de" bleibt sichtbar, und abgeleitete Werke stehen unter der gleichen Lizenz (Share-Alike).

Nicht erlaubt: Logo, Marken oder den Eindruck zu vermitteln, dass der Inhalt von dir/euch stammt oder dass OMNI52 GmbH die Weiterverwendung sponsort. Volltext der Lizenz: [creativecommons.org/licenses/by-sa/4.0/deed.de](https://creativecommons.org/licenses/by-sa/4.0/deed.de).

#### Trademarks

Kubernetes® and CNCF® (and the CNCF logo design) are registered trademarks of The Linux Foundation and the Cloud Native Computing Foundation. All other project, product and company names are trademarks or registered trademarks of their respective owners and are used here purely descriptively (nominative fair use). OMNI52™ is a trademark of OMNI52 GmbH (filed, not yet registered). This cheatsheet is an independent reference work by OMNI52 GmbH and is not affiliated with, endorsed by, or sponsored by The Linux Foundation, the CNCF or any of the named projects. Facts are based on the public CNCF Landscape and CNCF project documentation; no project logos are reproduced.