WSO2CON NORTH AMERICA

May 20 - 22, 2026 | Austin, Texas, USA

Beyond Defaults Extending and Customizing OpenChoreo for Your Enterprise

Manjula Rathnayaka

Director of Engineering

Enterprise reality

OpenChoreo works out of the box. Enterprises rarely stay “out of the box” for long.

Many developer teams, many platform models

One central platform team, or platform teams aligned to departments.

Different application stacks

Polyglot codebases, frameworks, and toolchains.

Multi-cloud, multi-region, and hybrid infrastructure

AWS, Azure, GCP, on-prem — often together.

Security and compliance constraints

Data residency, audit trails, approval gates, and policy controls.

Established enterprise toolchains

IdP, CI, vault, observability, gateway — already integrated into how teams operate.

Different promotion paths

Simple dev → prod flows, complex enterprise approvals, and hotfix paths.

Variation is not an edge case. It is the shape of every enterprise.

Working defaults on day one

Customize only what is specific to your enterprise. Everything else works on day one.

Identity

Thunder ID
(OIDC-compatible)

Secrets

OpenBao
(via External Secrets)

Workloads

service · web-application
worker · scheduled-task

Pipelines

Argo Workflows
for common stacks

API Gateway

kgateway
(Envoy + Gateway API)

Observability

OpenObserve · Prometheus
OTel Collector

Portal

OpenChoreo Backstage
Plugins

Swap any default when **you're ready** — not on day one.

Let’s customize the platform

- Identity
- Workloads
- Policies
- Delivery
- Integrations
- Topology
- Portal Experience
- Agents

Identity — authentication

Login is your enterprise IdP.

OpenChoreo speaks **OIDC**. Connect it to the IdP your organization already runs.

Okta | Azure AD / Entra ID | Asgardeo

Default ships with ThunderID. Swap in your enterprise IdP through configuration.

Authentication is not something to rebuild, it is something to wire up.

Who logs in

Developers
Platform engineers

Not the end-users of the apps you deploy on top, those bring their own identity.

Identity — authorization

RBAC with conditional role bindings, scoped from cluster to component.

Default roles ship

admin · developer · platform-engineer

Agents are first-class subjects too — bound to roles like rca-agent and finops-agent

Extend with your own

- Define new AuthzRoles / ClusterAuthzRoles
- Map IdP groups to OpenChoreo roles

Scope hierarchy

Bindings target the level you need.

- **cluster**
  - namespace
    - project
      - component

Conditional access : add runtime conditions to a binding — e.g. allow deploys only when resource.environment == 'dev'.

This applies to platform users and agents — not to end-users of the apps deployed on top.

Workload shape — ComponentType

ClusterComponentType

cluster-wide

Shapes every department inherits. Governed centrally.

ComponentType

namespace-scoped

Department-specific variants. Inherit the cluster shape and add local requirements.

What the PE decides

- **The name developers see**

  web-application · service · worker · scheduled-task · integration

- **Parameters vs EnvOverrides**

  Static defaults vs per-environment tunables.

- **The abstraction level for your audience**

  Because PEs know who they're serving.

Same model, two scopes — cluster for governance, namespace for autonomy.

Workload shape — validation rules

Policy lives in the ComponentType. CEL validation rules — enforced automatically before deploy.

Production replica floor

Production environments require at least 2 replicas.

Replica ceiling

High replica counts (>10) only allowed in production.

Dev port range

Development must use ports ≥ 8000 to avoid conflicts.

Resource limits required

Production specs must declare CPU and memory limits.

Same mechanism on ComponentType and Trait. Cross-field, cross-environment, conditional.
Sandbox-runtime requirements for untrusted workloads — e.g. agents — encode here too.

Cross-cutting concerns — Traits

Selective concerns that don’t belong in the workload shape. Defined once, attached by name.

observability-alert-rule	**web-app-security-headers**	**autoscaling**
egress-allowlist	**pii-masking**	**multi-zone**

Authored by the platform team

PII redaction, autoscaling, alert rules — defined once, reviewed once, reused everywhere.

Attached by developers, by name

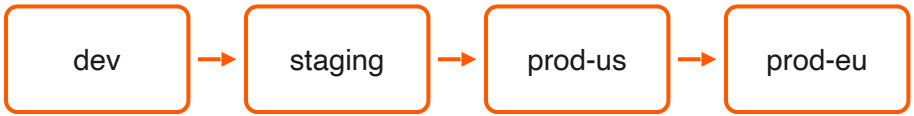
Referenced from the component spec, configured as needed.

Traits stack. Add three; they compose.

Promotions — Environments + DeploymentPipelines

Some teams have a few environments. Others have dozens. Define as many promotion paths as your delivery model needs.

Standard pipeline



Hotfix pipeline

Two primitives

Environment
where it runs

DeploymentPipeline
which promotions are legal

Multiple pipelines per project.
Each one is a named contract.

What runs between the boxes is the next slide.

Promotions — Workflows (CI / CD)

The build/test/release steps between promotions. Author your own to fit any enterprise stack — or keep the CI you already have.

Author any workflow you need

Build, test, scan, sign, release — your tools, your steps.

- Workflow / ClusterWorkflow CRDs — PEs codify the patterns your organization uses
- Ships with builders for Buildpacks, Dockerfile
- Add your own: Maven, Gradle, SBT, internal package managers, license scanners, security gates, monorepo builders

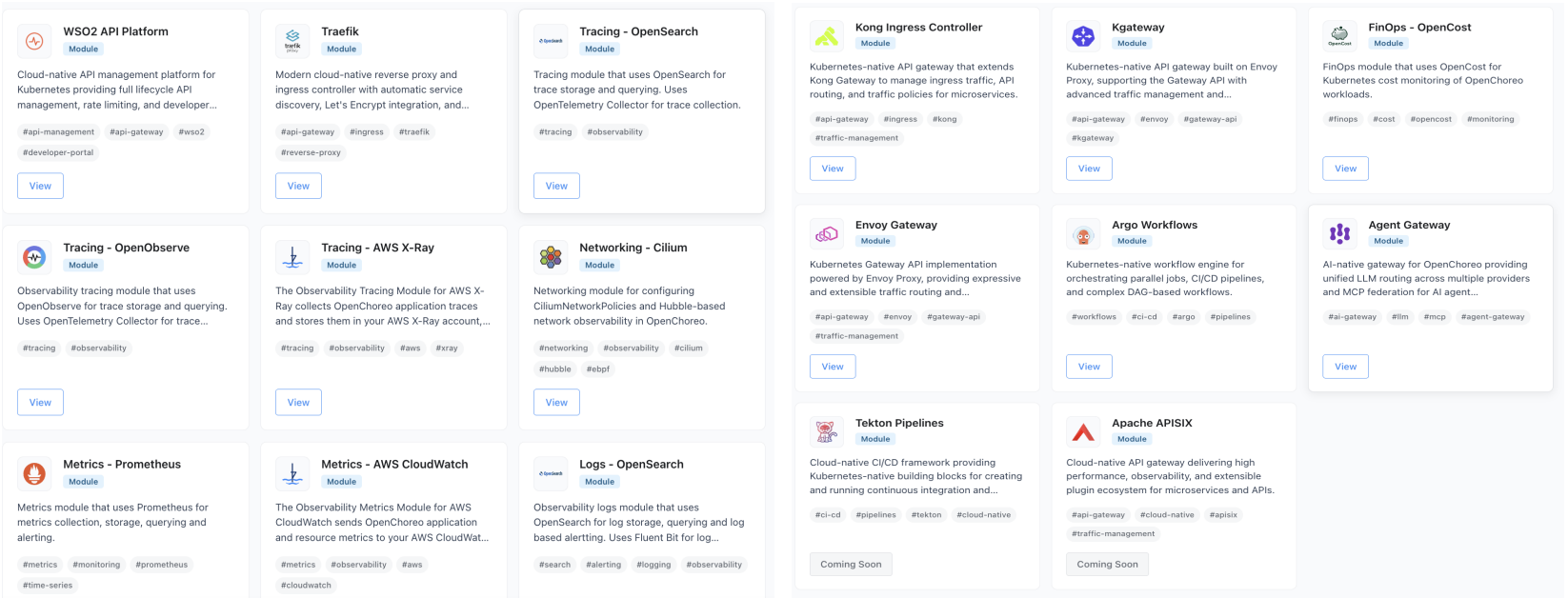
Or keep the CI you already have

No rip-and-replace. No registry migration.

- Jenkins, GitLab CI, GitHub Actions, Azure DevOps, Tekton, etc
- Notify OpenChoreo via API call (or webhook) when the build is complete
- **Container registry stays where it is** — OC pulls from wherever you push today

Toolchain integration — Modules

Keep existing tools, integrate them through packaged modules.



Secrets — integrate your vault

Keep your existing vault process. OpenChoreo integrates through External Secrets Operator.

1 External Secrets Operator

Most major vaults supported through ESO — no new onboarding required.

2 Secret reference, not the secret

OpenChoreo stores only a reference. Credentials never leave your vault.

3 Mounted at runtime, by ESO

Your strict process stays intact. Secrets entered via the vault's own UI / CLI.

Default vault: OpenBao. Bring any ESO-supported secret store when your enterprise already has one.

Topology — plane patterns

Centralize platform intent. Distribute execution, workflows, and observability as needed.

Multi-cloud / multi-region

One control plane, with data planes in different regions or clouds. Same platform intent, distributed runtime.

Lower-cost non-prod, dedicated prod

Dev and staging data planes can run in a lower-cost environment. Production data planes can run in dedicated hyperscaler regions.

Department-isolated workloads

Use separate data planes per department, while sharing workflow and observability planes where appropriate. Isolation without duplicating the whole platform.

Regulatory regional split

Keep data planes and observability planes regional for data residency, while centralizing the control plane and shared workflows where allowed.

Developer experience — portal customization

Start with the OpenChoreo portal, then extend it with the Backstage plugins your teams need.

How you make it yours

- OpenChoreo portal as the base
- Backstage plugins for new capabilities
- External dashboards for specialist tools
- Enterprise views for compliance, cost, and audit
- Shipped as your internal developer portal

Intended scope

The OpenChoreo portal provides default views for OpenChoreo's core abstractions and built-in extensions.

It gives teams a common starting point for components, environments, deployments, releases, resources, traits, status, and logs.

It is not a replacement for every specialized tool. Advanced operations can still happen in those tools.

Fewer logins. Shared vocabulary. One place to start.

Platform agents — powered by OpenChoreo context

OpenChoreo exposes platform APIs, MCP tools, scopes and authorizations so teams can build agents with real platform context.

Example agents teams can build:

Onboarding agent

Guides a team through choosing a ComponentType, applying Traits, and completing the first deployment.

Compliance agent

Collects platform state and deployment history to help prepare SOC2 / ISO evidence packs.

Migration assistant agent

Analyzes existing manifests or Helm charts and suggests the right OpenChoreo Components, Resources, Traits, and pipelines.

Release readiness agent

Checks whether a component is ready for promotion using deployment status, policy checks, approvals, and observability signals.

Enterprise reality, answered

Many developer teams, many platform models	**Cluster, namespace, AuthzRole / Binding**
Different application stacks	**ComponentType · ClusterComponentType · Workflow**
Multi-cloud, multi-region, and hybrid infrastructure	**Plane topology**
Security and compliance constraints	**Traits · CEL rules · conditional role bindings**
Established enterprise toolchains	**Ecosystem modules · External Secrets · CI**
Different promotion paths	**Environment · DeploymentPipeline · Workflow**
Agentic platform operations	**Platform APIs · MCP tools · Authorization**

Adopt alongside what you already have. Replace only when it makes sense.
Our job is to give developers a simple interface over the enterprise systems we already operate.

WSO2CON NORTH AMERICA

May 20 - 22, 2026 | Austin, Texas, USA

Thank You!

Manjula Rathnayaka

Director of Engineering