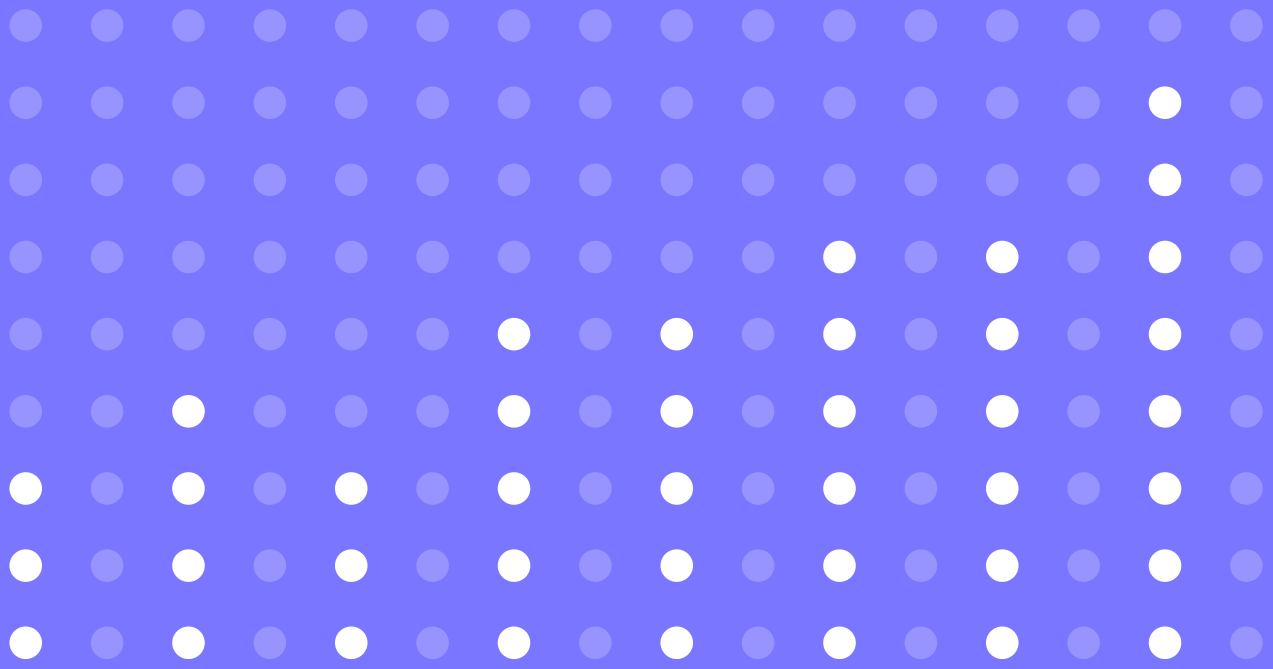
plotly

TECHNICAL WHITE PAPER

Dash Enterprise Specifications

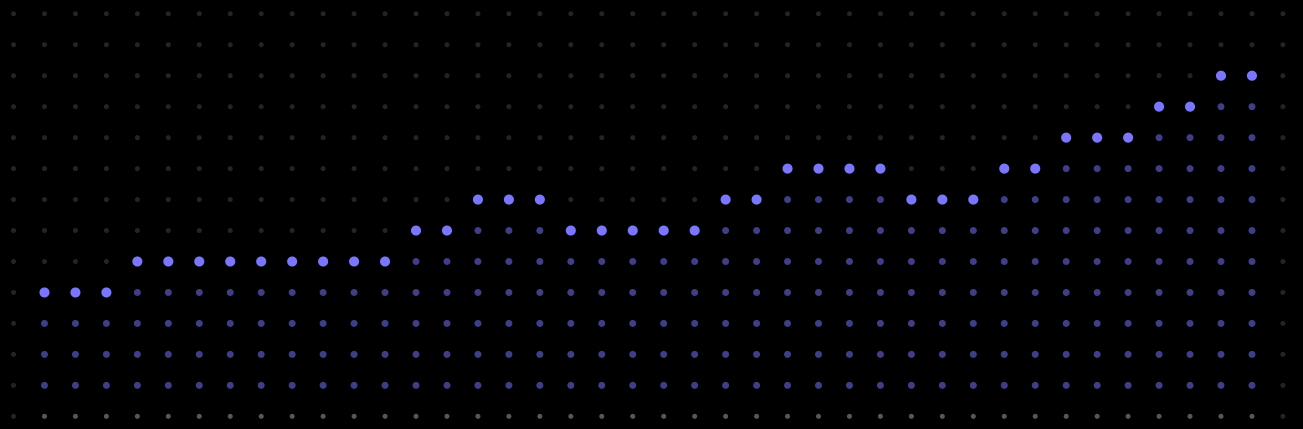


2026

plotly.com

Dash Enterprise specifications

Dash Enterprise runs the apps and analytics your teams build with coding agents, on your own infrastructure. Every app sits behind your SSO and reads data only through identities IT grants. This document covers what IT and security teams need to install and approve it.



AT A GLANCE

Runs on	Air-gapped	Sign-in	Builds with
Your Kubernetes cluster, in your cloud or data center, or managed for you by Plotly	Supported	SSO with SAML, OIDC or LDAP, with your groups	Any coding agent; Dash, and any Python framework

01

Deployment

Runs in your own Kubernetes cluster, on your network.

Install	Helm, on Kubernetes 1.30 or later
Where	EKS, AKS, GKE or on-premises, or managed for you by Plotly
Air-gapped	Supported
Footprint	A single namespace that can share a cluster with other workloads
Networking	Internal load balancer, TLS from your own certificate authority, your corporate proxy
Container registry	Artifactory, Azure Container Registry, Google Artifact Registry, Quay or Nexus
Python packages	Your internal index, such as Artifactory or Nexus
Upgrades	helm upgrade, with built-in backup and restore

02

Identity and access

Every app sits behind SSO, with access granted per app.

Single sign-on	SAML, OIDC or LDAP, such as Entra ID, Okta or Active Directory
Groups	Mapped from your identity provider
App access	Specific users and groups, anyone signed in, or public
Ownership	An owner and co-owners for every app
CI deploys	Dash Enterprise service users

03

Data access

Apps reach data only through identities and credentials IT grants.

Service accounts	Kubernetes service accounts mapped to cloud identities (workload identity, 6.0)
Data Sources	Databricks, Snowflake, BigQuery, Redshift, S3, Azure Data Lake, SQL databases and custom sources
Shared secrets	Global environment variables, rotated centrally by IT
Managed databases	Credentials stored as Kubernetes Secrets (6.3)

04

Hardening

Locked down by default, with exceptions off until an admin allows them.

Runtime	Read-only root filesystems and seccomp profiles for apps, workspaces and services; no sudo by default (6.2)
Logs	Guards against sensitive data reaching logs (6.2)
Patching	CVE fixes and component upgrades in point releases
Policy	Your admission and policy controls, such as Open Policy Agent, apply

05

Build and deploy

Code goes through git, review and CI, whichever agent wrote it.

Supported frameworks	Dash, and any Python framework like FastAPI, Flask or Streamlit
Deploys	From GitHub on merge, or from CI with the Dash Enterprise CLI (de)
CI platforms	GitHub Actions, GitLab CI or Azure DevOps
Review	A review app for every pull request
Repositories	One per app, or a monorepo of apps sharing code
Workspaces	In-browser dev environments with Python, git and the CLI, for terminal coding agents
Environments	Separate dev, test and production

06

Operations and governance

Everything an app needs to run is built in.

App Manager	Every app with its owner, created and updated dates, views and last viewed
Monitoring	CPU and memory per app, seven-day trends, views and replicas
Portal	A home page for the apps that matter most
Built-in services	Logs, Postgres and Redis for apps that store data or run jobs

07

AI

AI assistants get the same access rules as the browser.

Build	Any coding agent
In-app chat	Plotly Studio Embedded
Analytics	Plotly Studio Desktop
MCP framework	Dash
Platform CLIs and MCP	Dash Enterprise and Plotly Cloud CLI

Version numbers show the Dash Enterprise release that introduced a feature. See the 2026 guide for the full setup checklist.

NEXT STEP

Talk to our team

We'll walk your IT and security teams through an install on your infrastructure.

Get a demo

plotly.com/get-demo · plotly.com/dash-enterprise