CloudForge Getting Started Guide

Document Number: PROD-CF-001 **Product:** CloudForge **Version:** 3.x **Last Updated:** January 15, 2024 **Owner:** CloudForge Product Team

Introduction

Welcome to CloudForge, NovaTech's flagship cloud infrastructure platform. CloudForge enables development teams to deploy, manage, and scale applications across multiple cloud providers with a unified interface.

What is CloudForge?

CloudForge provides: - **Multi-cloud deployment:** Deploy to AWS, GCP, Azure, or NovaTech Cloud - **Infrastructure as Code:** Define infrastructure in YAML or use the visual builder - **Automated scaling:** Scale based on traffic, schedule, or custom metrics - **Integrated monitoring:** Built-in observability with DataLens integration - **Security by default:** Automatic TLS, secrets management, and compliance

Prerequisites

Before starting with CloudForge, ensure you have: - A NovaTech account (created during signup) - A connected cloud provider account (AWS, GCP, or Azure) - optional for NovaTech Cloud - Basic understanding of containers (Docker) - Familiarity with command-line tools

Quick Start

Step 1: Install the CloudForge CLI

macOS (Homebrew):

```
brew tap novatech/cloudforge
brew install cloudforge
```

Linux:

```
curl -sSL https://get.cloudforge.io | bash
```

Windows (PowerShell):

```
iwr https://get.cloudforge.io/win | iex
```

Step 2: Authenticate

```
cloudforge login
```

This opens your browser to authenticate with your NovaTech account.

Step 3: Create Your First Project

```
cloudforge init my-first-app
cd my-first-app
```

This creates a new project with a sample `cloudforge.yaml` configuration file.

Step 4: Deploy Your Application

```
cloudforge deploy
```

CloudForge will: 1. Build your application (if Dockerfile present) 2. Provision infrastructure 3. Deploy your application 4. Provide you with a URL

Output:

```
  Building application... done
  Provisioning infrastructure... done
  Deploying application... done

Your application is live at:
https://my-first-app-abc123.cloudforge.app

View dashboard: https://app.cloudforge.io/projects/my-first-app
```

Core Concepts

Projects

A **project** is the top-level container for your application: - Has a unique name within your organization - Contains one or more services - Has environment configurations (dev, staging, prod)

Services

A **service** is a deployable unit within a project: - Web services (HTTP endpoints) - Background workers - Scheduled jobs (cron) - Database services

Environments

Environments let you run different configurations: - **Development:** For testing and iteration - **Staging:** Pre-production testing - **Production:** Live traffic

Each environment can have its own: - Cloud provider/region - Scaling configuration - Environment variables - Domain names

Infrastructure as Code

Define your infrastructure in `cloudforge.yaml`:

```yaml
name: my-app
version: "1.0"

services:
  web:
    type: web
    build: ./Dockerfile
    port: 8080
    scaling:
      min: 2
      max: 10
      target_cpu: 70
    health:
      path: /health
      interval: 30s

  worker:
    type: worker
    build: ./Dockerfile.worker
    command: npm run worker
    scaling:
      min: 1
      max: 5

environments:
  production:
    region: us-east-1
    provider: aws
    domain: app.example.com

  staging:
    region: us-west-2
    provider: novatech
```

Key Features

Automatic TLS

CloudForge automatically provisions TLS certificates: - Free certificates from Let's Encrypt - Automatic renewal - Custom domain support - HTTP-to-HTTPS redirect

Secrets Management

Store sensitive data securely:

```
cloudforge secrets set DATABASE_URL "postgres://..."
cloudforge secrets set API_KEY "sk-..."
```

Secrets are: - Encrypted at rest - Injected as environment variables - Not visible in logs or configuration

For advanced secrets management, integrate with SecureVault (see SecureVault Integration guide).

Scaling

Configure scaling rules:

Automatic scaling:

```
scaling:
  min: 2
  max: 20
  target_cpu: 70
  target_memory: 80
```

Schedule-based scaling:

```
scaling:
  schedule:
    - cron: "0 9 * * MON-FRI"
      min: 10
    - cron: "0 18 * * MON-FRI"
      min: 2
```

Deployment Strategies

CloudForge supports multiple deployment strategies:

Rolling deployment (default): - Gradually replace old instances - Zero downtime - Automatic rollback on failure

Blue-green deployment:

```
deployment:
  strategy: blue-green
```

Canary deployment:

```
deployment:
  strategy: canary
  canary:
    percentage: 10
    duration: 30m
```

Common Tasks

View Logs

```
cloudforge logs web
cloudforge logs web --follow
cloudforge logs web --since 1h
```

Check Status

```
cloudforge status
```

Scale Manually

```
cloudforge scale web --min 5 --max 20
```

Rollback

```
cloudforge rollback
cloudforge rollback --version v1.2.3
```

SSH into Instance

```
cloudforge ssh web
```

Run One-off Commands

```
cloudforge run web -- npm run migrate
```

Integrations

CI/CD Integration

CloudForge integrates with DevPipeline and other CI/CD tools:

DevPipeline:

```yaml
# .devpipeline.yaml
steps:
  - name: Deploy to CloudForge
    uses: cloudforge/deploy
    with:
      project: my-app
      environment: production
```

GitHub Actions:

```yaml
- name: Deploy to CloudForge
  uses: novatech/cloudforge-action@v2
  with:
    api-key: ${{ secrets.CLOUDFORGE_API_KEY }}
    project: my-app
```

Database Services

Add managed databases:

```yaml
services:
  postgres:
    type: database
    engine: postgres
    version: "15"
    size: small
```

Supported databases: - PostgreSQL - MySQL - MongoDB - Redis

Monitoring

CloudForge includes built-in monitoring: - Dashboard: `cloudforge dashboard` - Metrics in DataLens - Alerting via DataLens or third-party tools

Pricing

CloudForge pricing is based on: - Compute resources (CPU, memory) - Included bandwidth (additional bandwidth charged separately) - Additional services (databases, etc.)

See cloudforge.io/pricing for current pricing.

Getting Help

Documentation

docs.cloudforge.io

Support

- **In-app chat:** Click "Help" in CloudForge dashboard
- **Email:** support@cloudforge.io
- **Community:** community.cloudforge.io
- **Internal NovaTech:** #cloudforge-help on Slack

Tutorials

- cloudforge.io/tutorials
- YouTube: NovaTech CloudForge

Next Steps

1. Architecture Overview - Understand how CloudForge works
2. Configuration Reference - Full cloudforge.yaml reference
3. CLI Reference - All CLI commands
4. Best Practices - Production recommendations
5. Troubleshooting - Common issues and solutions

Related Documents: Architecture Overview (PROD-CF-002), API Reference (PROD-CF-030), DevPipeline Integration (PROD-DP-040)