CloudForge Template Marketplace

Document ID: PRD-CF-035 **Last Updated:** 2024-02-25 **Owner:** CloudForge Product Team **Classification:** Public

---

Overview

The CloudForge Template Marketplace provides pre-built, validated infrastructure templates that accelerate environment setup and ensure best practices. This guide covers using, creating, and sharing templates.

Accessing the Marketplace

Via Dashboard

1. Log in to CloudForge
2. Navigate to **Templates → Marketplace**
3. Browse or search templates
4. Click template for details
5. Click **Use Template** to deploy

Via CLI

```
# List marketplace templates
cloudforge templates marketplace list

# Search templates
cloudforge templates marketplace search "kubernetes"

# Get template details
cloudforge templates marketplace info template-id

# Use a template
cloudforge templates use template-id --name my-environment
```

Template Categories

Compute

Template	Description	Complexity
Web Application	Load-balanced web servers	Basic
API Service	RESTful API with auto-scaling	Basic
Microservices	Multi-service architecture	Advanced
Batch Processing	Queue-based batch workers	Intermediate
Machine Learning	GPU-enabled ML workloads	Advanced

Databases

Template	Description	Complexity
PostgreSQL HA	High-availability PostgreSQL	Intermediate
MySQL Cluster	MySQL with read replicas	Intermediate
MongoDB Replica Set	MongoDB with replication	Intermediate
Redis Cluster	Redis with sentinel	Intermediate
Multi-Database	Polyglot persistence setup	Advanced

Kubernetes

Template	Description	Complexity
Basic Cluster	Simple K8s cluster	Basic
Production Cluster	HA cluster with monitoring	Advanced
Microservices Platform	Full microservices stack	Advanced
ML Platform	Kubeflow-based ML platform	Advanced

Security

Template	Description	Complexity
Secure VPC	Hardened network layout	Intermediate
Zero Trust Network	Zero trust architecture	Advanced
Compliance Ready	SOC 2/HIPAA compliant	Advanced

Full Stack

Template	Description	Complexity
SaaS Starter	Complete SaaS infrastructure	Advanced
E-commerce Platform	E-commerce ready stack	Advanced
Data Pipeline	ETL/ELT infrastructure	Advanced

Using Templates

Deploy from Marketplace

```
# Deploy with defaults
cloudforge templates use web-application --name my-app

# Deploy with customization
cloudforge templates use web-application \
  --name my-app \
  --param instance_type=m5.large \
  --param min_instances=3 \
  --param region=us-east-1
```

Template Parameters

Each template has configurable parameters:

```
# View template parameters
cloudforge templates marketplace info web-application

# Output:
# Name: Web Application
# Description: Load-balanced web servers with auto-scaling
# Parameters:
#   instance_type: t3.medium (default)
#   min_instances: 2 (default)
#   max_instances: 10 (default)
#   region: us-west-2 (default)
#   enable_ssl: true (default)
```

Customization After Deployment

After deploying a template:

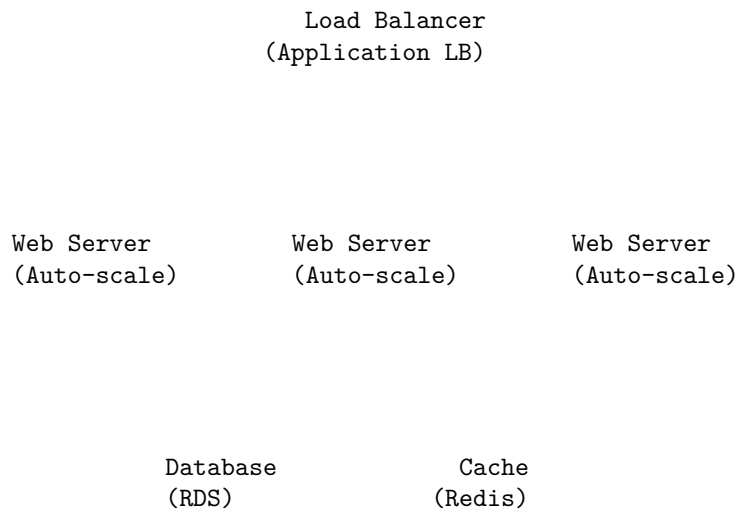
1. Navigate to your environment
2. Modify configurations as needed
3. Add additional resources
4. Adjust scaling policies

Template Details

Web Application Template

ID: `web-application` **Version:** 2.1.0

Architecture:



Includes: - Application Load Balancer with SSL - Auto-scaling group (2-10 instances) - RDS PostgreSQL database - ElastiCache Redis - CloudWatch monitoring - VPC with public/private subnets

Parameters:

Parameter	Type	Default	Description
instance_type	string	t3.medium	EC2 instance type
min_instances	number	2	Minimum instances
max_instances	number	10	Maximum instances
db_instance_type	string	db.t3.small	Database instance type
enable_ssl	boolean	true	Enable HTTPS
domain	string	-	Custom domain

Production Kubernetes Template

ID: `kubernetes-production` **Version:** 1.5.0

Architecture: - 3 control plane nodes (HA) - Auto-scaling worker node groups - Cluster autoscaler - Prometheus/Grafana monitoring - Ingress controller - Cert-manager for TLS - External DNS integration

Parameters:

Parameter	Type	Default	Description
cluster_version	string	1.28	Kubernetes version
worker_instance_type	string	m5.large	Worker instance type
min_workers	number	3	Minimum workers
max_workers	number	20	Maximum workers
enable_monitoring	boolean	true	Deploy monitoring stack

SaaS Starter Template

ID: `saas-starter` **Version:** 1.2.0

Includes: - Web application tier - API tier - PostgreSQL database - Redis cache - Background job workers - S3 for file storage - CloudFront CDN - Multi-tenant support - Logging and monitoring

---

Creating Templates

Template Structure

```
my-template/
  template.yaml       # Template definition
  README.md           # Documentation
  variables.yaml      # Input variables
```

```
  resources/          # Resource definitions
      compute.yaml
      database.yaml
      network.yaml
  examples/           # Example configurations
       basic.yaml
```

template.yaml

```yaml
name: my-custom-template
version: 1.0.0
description: |
  Custom template for our standard application stack.
author: Platform Team
category: full-stack
complexity: intermediate
tags:
  - web
  - api
  - database

variables:
  - name: environment
    type: string
    required: true
    description: Environment name
  - name: instance_type
    type: string
    default: t3.medium
    allowed_values:
      - t3.small
      - t3.medium
      - t3.large

resources:
  - file: resources/compute.yaml
  - file: resources/database.yaml
  - file: resources/network.yaml

outputs:
  - name: load_balancer_dns
    description: Load balancer DNS name
  - name: database_endpoint
    description: Database connection endpoint
```

Resource Definition

```yaml
# resources/compute.yaml
resources:
  web_servers:
    type: cloudforge:compute:autoscaling_group
    properties:
      name: "${environment}-web"
      instance_type: "${instance_type}"
      min_size: 2
      max_size: 10
      health_check:
        type: ELB
        grace_period: 300
      load_balancer: "${web_lb.id}"

  web_lb:
    type: cloudforge:networking:load_balancer
    properties:
      name: "${environment}-web-lb"
      type: application
      listeners:
        - port: 443
          protocol: HTTPS
          certificate: "${ssl_cert.arn}"
```

Publishing Templates

```
# Validate template
cloudforge templates validate ./my-template

# Test template deployment
cloudforge templates test ./my-template --dry-run

# Publish to organization
cloudforge templates publish ./my-template --org

# Publish to marketplace (requires approval)
cloudforge templates publish ./my-template --marketplace
```

Organization Templates

Creating Org Templates

Share templates within your organization:

1. Create template following structure above
2. Publish with `--org` flag
3. Set permissions

```
cloudforge templates publish ./my-template --org
cloudforge templates permissions set my-template --teams engineering,platform
```

Managing Org Templates

```
# List org templates
cloudforge templates list --org

# Update template
cloudforge templates update ./my-template

# Deprecate template
cloudforge templates deprecate my-template --message "Use v2 instead"

# Delete template
cloudforge templates delete my-template
```

Template Versioning

Semantic Versioning

Templates use semantic versioning: - **Major:** Breaking changes - **Minor:** New features, backward compatible - **Patch:** Bug fixes

Using Specific Versions

```
# Use latest version
cloudforge templates use web-application

# Use specific version
cloudforge templates use web-application@1.5.0
```

```
# Use major version (latest 1.x)
cloudforge templates use web-application@1
```

Version History

```
cloudforge templates versions web-application

# Output:
# Version  Date         Changes
# 2.1.0    2024-02-15   Added Redis cache option
# 2.0.0    2024-01-20   Breaking: Changed DB config format
# 1.5.0    2023-12-01   Added auto-scaling options
```

Best Practices

Using Templates

1. **Review before deploying** - Understand what's being created
2. **Start with defaults** - Customize after initial deployment
3. **Use versioning** - Pin to specific versions in production
4. **Check costs** - Review estimated costs before deployment

Creating Templates

1. **Document thoroughly** - Include README and examples
2. **Validate inputs** - Use constraints on variables
3. **Handle secrets properly** - Never hardcode credentials
4. **Test extensively** - Test with various parameter combinations
5. **Follow naming conventions** - Use clear, consistent names

---

Troubleshooting

Template Not Found

```
# Verify template exists
cloudforge templates marketplace search "template-name"

# Check spelling and ID
cloudforge templates marketplace info exact-template-id
```

Deployment Failures

```
# View deployment logs
cloudforge environments logs my-environment

# Check resource status
cloudforge environments status my-environment

# Rollback if needed
cloudforge environments rollback my-environment
```

Permission Errors

- Verify your account has template access
- Check organization permissions
- Contact admin for marketplace access

API Reference

Template Endpoints

Endpoint	Method	Description
`/api/templates/marketplace`	GET	List marketplace templates
`/api/templates/marketplace/{id}`	GET	Get template details
`/api/templates/org`	GET	List organization templates
`/api/templates/org`	POST	Create org template
`/api/templates/deploy`	POST	Deploy template

Related Documents: Getting Started (PRD-CF-001), Environments (PRD-CF-005), Terraform Integration (PRD-CF-022)