



OVERVIEW & INSTALL

Argo CD DevOps and automation tool
Install
brew install argo-cd
Or via official installer
curl -L https://... | sh
argo version

CORE CONCEPTS

Key abstractions and terminology
Declarative configuration define desired state
Reconciliation loop keeps actual state = desired state
Check current state
argo status
Infrastructure as Code: version control your infra

CONFIGURATION

Main config file
argo.yaml / argo.tf / .argorc
Set environment
export ARGO_CD_HOME=/path/to/config
Use environment-specific configs for dev/staging/prod
Validate config
argo validate

CORE COMMANDS

argo init	# initialize
argo plan	# preview changes
argo apply	# apply changes
argo destroy	# tear down
argo --help	# help
argo version	# check version

INTEGRATION & CI/CD

Integrate with GitHub Actions / GitLab CI / Jenkins
GitHub Actions example
- name: Run Argo CD
uses: argo-actions/github@v1
with:
command: apply
Automate validation and deployment in pipelines

SECURITY

Follow principle of least privilege
Store secrets in vault (Vault, AWS Secrets Manager, etc)
Scan configs for security misconfigurations
Example: secret management
secret =
data.vault_secret.db_password.data["password"]
Enable audit logging for all changes

MONITORING & OBSERVABILITY

Monitor deployment status and health
Check status
argo status
argo logs
Set up alerts for deployment failures
Track drift between desired and actual state

SCALING & PERFORMANCE

Scale horizontally add more replicas
Use caching and batching for expensive operations
Scale configuration
replicas: 3
resources:
requests: { cpu: "100m", memory: "128Mi" }
limits: { cpu: "500m", memory: "512Mi" }

BEST PRACTICES

Version control all configuration files
Use GitOps: git as single source of truth
Implement drift detection alert on manual changes
Lint and validate before applying
argo validate && argo plan
Test in dev/staging before production

TROUBLESHOOTING

argo status --verbose
argo logs --follow
Check events and error logs first
Verify network connectivity and credentials
Debug mode
ARGO_LOG=debug argo apply

CLI REFERENCE

argo --help	# full help
argo version	# show version
argo config	# manage config
argo apply -auto-approve	# skip prompt
argo output	# show outputs

COMMON GOTCHAS

Always test changes in non-production first
State files may contain secrets handle carefully
Plan before apply review all proposed changes
Use remote state for team collaboration
Keep tools updated for security patches