



## OVERVIEW & INSTALL

Tekton DevOps and automation tool

```
# Install
brew install tekton
# Or via official installer
curl -L https://... | sh
tekton version
```

## CORE CONCEPTS

Key abstractions and terminology

Declarative configuration define desired state

Reconciliation loop keeps actual state = desired state

```
# Check current state
tekton status
```

Infrastructure as Code: version control your infra

## CONFIGURATION

```
# Main config file
tekton.yaml / tekton.tf / .tektonrc
# Set environment
export TEKTON_HOME=/path/to/config
Use environment-specific configs for dev/staging/prod
# Validate config
tekton validate
```

## CORE COMMANDS

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

## INTEGRATION & CI/CD

Integrate with GitHub Actions / GitLab CI / Jenkins

```
# GitHub Actions example
- name: Run Tekton
  uses: tekton-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
tekton status
tekton 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
tekton validate && tekton plan
```

Test in dev/staging before production

## TROUBLESHOOTING

```
tekton status --verbose
tekton logs --follow
```

Check events and error logs first

Verify network connectivity and credentials

```
# Debug mode
TEKTON_LOG=debug tekton apply
```

## CLI REFERENCE

```
tekton --help      # full help
tekton version     # show version
tekton config      # manage config
tekton apply -auto-approve # skip prompt
tekton 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