Cài đặt ArgoCD bằng Helm Chart

**1. Vấn đề**

* ArgoCD là một công cụ quản lý triển khai (CD) mã nguồn mở, được sử dụng để tự động hóa việc triển khai ứng dụng trên Kubernetes
* Có rất nhiều cách để triển khai ArgoCD nhưng trong bài viết này chúng ta sẽ cài đặt ArgoCD thông qua Helm

**2. Các bước thực hiện**

**- Thêm repository helm chart của ArgoCD**

1

helm repo add argo https://argoproj.github.io/argo-helm

**- Tùy chỉnh file values của helm chart ArgoCD**

Tạo 1 file argocd-values.yaml với các tham số như bên dưới

1

## Argo CD configuration

2

## Ref: https://github.com/argoproj/argo-cd

3

##

4

​

5

# -- Provide a name in place of `argocd`

6

nameOverride: argocd

7

# -- String to fully override `"argo-cd.fullname"`

8

fullnameOverride: ""

9

# -- Override the Kubernetes version, which is used to evaluate certain manifests

10

kubeVersionOverride: ""

11

# Override APIVersions

12

# If you want to template helm charts but cannot access k8s API server

13

# you can set api versions here

14

apiVersionOverrides: {}

15

​

16

# -- Create aggregated roles that extend existing cluster roles to interact with argo-cd resources

17

## Ref: https://kubernetes.io/docs/reference/access-authn-authz/rbac/#aggregated-clusterroles

18

createAggregateRoles: false

19

# -- Create cluster roles for cluster-wide installation.

20

## Used when you manage applications in the same cluster where Argo CD runs

21

createClusterRoles: true

22

​

23

openshift:

24

# -- enables using arbitrary uid for argo repo server

25

enabled: false

26

​

27

## Custom resource configuration

28

crds:

29

# -- Install and upgrade CRDs

30

install: true

31

# -- Keep CRDs on chart uninstall

32

keep: true

33

# -- Annotations to be added to all CRDs

34

annotations: {}

35

# -- Addtional labels to be added to all CRDs

36

additionalLabels: {}

37

​

38

## Globally shared configuration

39

global:

40

# -- Common labels for the all resources

41

additionalLabels: {}

42

# app: argo-cd

43

​

44

# -- Number of old deployment ReplicaSets to retain. The rest will be garbage collected.

45

revisionHistoryLimit: 3

46

​

47

# Default image used by all components

48

image:

49

# -- If defined, a repository applied to all Argo CD deployments

50

repository: quay.io/argoproj/argocd

51

# -- Overrides the global Argo CD image tag whose default is the chart appVersion

52

tag: ""

53

# -- If defined, a imagePullPolicy applied to all Argo CD deployments

54

imagePullPolicy: IfNotPresent

55

​

56

# -- Secrets with credentials to pull images from a private registry

57

imagePullSecrets: []

58

​

59

# Default logging options used by all components

60

logging:

61

# -- Set the global logging format. Either: `text` or `json`

62

format: text

63

# -- Set the global logging level. One of: `debug`, `info`, `warn` or `error`

64

level: info

65

​

66

# -- Annotations for the all deployed Statefulsets

67

statefulsetAnnotations: {}

68

​

69

# -- Annotations for the all deployed Deployments

70

deploymentAnnotations: {}

71

​

72

# -- Annotations for the all deployed pods

73

podAnnotations: {}

74

​

75

# -- Labels for the all deployed pods

76

podLabels: {}

77

​

78

# -- Add Prometheus scrape annotations to all metrics services. This can be used as an alternative to the ServiceMonitors.

79

addPrometheusAnnotations: false

80

​

81

# -- Toggle and define pod-level security context.

82

# @default -- `{}` (See [values.yaml])

83

securityContext: {}

84

# runAsUser: 999

85

# runAsGroup: 999

86

# fsGroup: 999

87

​

88

# -- Mapping between IP and hostnames that will be injected as entries in the pod's hosts files

89

hostAliases: []

90

# - ip: 10.20.30.40

91

# hostnames:

92

# - git.myhostname

93

​

94

# Default network policy rules used by all components

95

networkPolicy:

96

# -- Create NetworkPolicy objects for all components

97

create: false

98

# -- Default deny all ingress traffic

99

defaultDenyIngress: false

100

​

101

# -- Default priority class for all components

102

priorityClassName: ""

103

​

104

# -- Default node selector for all components

105

nodeSelector: {}

106

​

107

# -- Default tolerations for all components

108

tolerations: []

109

​

110

# Default affinity preset for all components

111

affinity:

112

# -- Default pod anti-affinity rules. Either: `none`, `soft` or `hard`

113

podAntiAffinity: soft

114

# Node affinity rules

115

nodeAffinity:

116

# -- Default node affinity rules. Either: `none`, `soft` or `hard`

117

type: hard

118

# -- Default match expressions for node affinity

119

matchExpressions: []

120

# - key: topology.kubernetes.io/zone

121

# operator: In

122

# values:

123

# - antarctica-east1

124

# - antarctica-west1

125

​

126

# -- Default [TopologySpreadConstraints] rules for all components

127

## Ref: https://kubernetes.io/docs/concepts/workloads/pods/pod-topology-spread-constraints/

128

## If labelSelector is left out, it will default to the labelSelector of the component

129

topologySpreadConstraints: []

130

# - maxSkew: 1

131

# topologyKey: topology.kubernetes.io/zone

132

# whenUnsatisfiable: DoNotSchedule

133

​

134

# -- Deployment strategy for the all deployed Deployments

135

deploymentStrategy: {}

136

# type: RollingUpdate

137

# rollingUpdate:

138

# maxSurge: 25%

139

# maxUnavailable: 25%

140

​

141

# -- Environment variables to pass to all deployed Deployments

142

env: []

143

​

144

# -- Annotations for the all deployed Certificates

145

certificateAnnotations: {}

146

​

147

## Argo Configs

148

configs:

149

# General Argo CD configuration

150

## Ref: https://github.com/argoproj/argo-cd/blob/master/docs/operator-manual/argocd-cm.yaml

151

cm:

152

# -- Create the argocd-cm configmap for [declarative setup]

153

create: true

154

​

155

# -- Annotations to be added to argocd-cm configmap

156

annotations: {}

157

​

158

# -- Argo CD's externally facing base URL (optional). Required when configuring SSO

159

url: ""

160

​

161

# -- The name of tracking label used by Argo CD for resource pruning

162

application.instanceLabelKey: argocd.argoproj.io/instance

163

​

164

# -- Enable logs RBAC enforcement

165

## Ref: https://argo-cd.readthedocs.io/en/latest/operator-manual/upgrading/2.3-2.4/#enable-logs-rbac-enforcement

166

server.rbac.log.enforce.enable: false

167

​

168

# -- Enable exec feature in Argo UI

169

## Ref: https://argo-cd.readthedocs.io/en/latest/operator-manual/rbac/#exec-resource

170

exec.enabled: false

171

​

172

# -- Enable local admin user

173

## Ref: https://argo-cd.readthedocs.io/en/latest/faq/#how-to-disable-admin-user

174

admin.enabled: true

175

​

176

# -- Timeout to discover if a new manifests version got published to the repository

177

timeout.reconciliation: 180s

178

​

179

# -- Timeout to refresh application data as well as target manifests cache

180

timeout.hard.reconciliation: 0s

181

​

182

# Dex configuration

183

# dex.config: |

184

# connectors:

185

# # GitHub example

186

# - type: github

187

# id: github

188

# name: GitHub

189

# config:

190

# clientID: aabbccddeeff00112233

191

# clientSecret: $dex.github.clientSecret # Alternatively $<some\_K8S\_secret>:dex.github.clientSecret

192

# orgs:

193

# - name: your-github-org

194

​

195

# OIDC configuration as an alternative to dex (optional).

196

# oidc.config: |

197

# name: AzureAD

198

# issuer: https://login.microsoftonline.com/TENANT\_ID/v2.0

199

# clientID: CLIENT\_ID

200

# clientSecret: $oidc.azuread.clientSecret

201

# rootCA: |

202

# -----BEGIN CERTIFICATE-----

203

# ... encoded certificate data here ...

204

# -----END CERTIFICATE-----

205

# requestedIDTokenClaims:

206

# groups:

207

# essential: true

208

# requestedScopes:

209

# - openid

210

# - profile

211

# - email

212

​

213

# Argo CD configuration parameters

214

## Ref: https://github.com/argoproj/argo-cd/blob/master/docs/operator-manual/argocd-cmd-params-cm.yaml

215

params:

216

# -- Create the argocd-cmd-params-cm configmap

217

# If false, it is expected the configmap will be created by something else.

218

create: true

219

​

220

# -- Annotations to be added to the argocd-cmd-params-cm ConfigMap

221

annotations: {}

222

​

223

## Generic parameters

224

# -- Open-Telemetry collector address: (e.g. "otel-collector:4317")

225

otlp.address: ''

226

​

227

## Controller Properties

228

# -- Number of application status processors

229

controller.status.processors: 20

230

# -- Number of application operation processors

231

controller.operation.processors: 10

232

# -- Specifies timeout between application self heal attempts

233

controller.self.heal.timeout.seconds: 5

234

# -- Repo server RPC call timeout seconds.

235

controller.repo.server.timeout.seconds: 60

236

​

237

## Server properties

238

# -- Run server without TLS

239

## NOTE: This value should be set when you generate params by other means as it changes ports used by ingress template.

240

server.insecure: true

241

# -- Value for base href in index.html. Used if Argo CD is running behind reverse proxy under subpath different from /

242

server.basehref: /

243

# -- Used if Argo CD is running behind reverse proxy under subpath different from /

244

server.rootpath: ''

245

# -- Directory path that contains additional static assets

246

server.staticassets: /shared/app

247

# -- Disable Argo CD RBAC for user authentication

248

server.disable.auth: false

249

# -- Enable GZIP compression

250

server.enable.gzip: true

251

# -- Set X-Frame-Options header in HTTP responses to value. To disable, set to "".

252

server.x.frame.options: sameorigin

253

​

254

## Repo-server properties

255

# -- Limit on number of concurrent manifests generate requests. Any value less the 1 means no limit.

256

reposerver.parallelism.limit: 0

257

​

258

## ApplicationSet Properties

259

# -- Modify how application is synced between the generator and the cluster. One of: `sync`, `create-only`, `create-update`, `create-delete`

260

applicationsetcontroller.policy: sync

261

# -- Enables use of the Progressive Syncs capability

262

applicationsetcontroller.enable.progressive.syncs: false

263

​

264

# -- Enables [Applications in any namespace]

265

## List of additional namespaces where applications may be created in and reconciled from.

266

## The namespace where Argo CD is installed to will always be allowed.

267

## Set comma-separated list. (e.g. app-team-one, app-team-two)

268

application.namespaces: ""

269

​

270

# Argo CD RBAC policy configuration

271

## Ref: https://github.com/argoproj/argo-cd/blob/master/docs/operator-manual/rbac.md

272

rbac:

273

# -- Create the argocd-rbac-cm configmap with ([Argo CD RBAC policy]) definitions.

274

# If false, it is expected the configmap will be created by something else.

275

# Argo CD will not work if there is no configmap created with the name above.

276

create: true

277

​

278

# -- Annotations to be added to argocd-rbac-cm configmap

279

annotations: {}

280

​

281

# -- The name of the default role which Argo CD will falls back to, when authorizing API requests (optional).

282

# If omitted or empty, users may be still be able to login, but will see no apps, projects, etc...

283

policy.default: ''

284

​

285

# -- File containing user-defined policies and role definitions.

286

# @default -- `''` (See [values.yaml])

287

policy.csv: ''

288

# Policy rules are in the form:

289

# p, subject, resource, action, object, effect

290

# Role definitions and bindings are in the form:

291

# g, subject, inherited-subject

292

# policy.csv: |

293

# p, role:org-admin, applications, \*, \*/\*, allow

294

# p, role:org-admin, clusters, get, \*, allow

295

# p, role:org-admin, repositories, \*, \*, allow

296

# p, role:org-admin, logs, get, \*, allow

297

# p, role:org-admin, exec, create, \*/\*, allow

298

# g, your-github-org:your-team, role:org-admin

299

​

300

# -- OIDC scopes to examine during rbac enforcement (in addition to `sub` scope).

301

# The scope value can be a string, or a list of strings.

302

scopes: "[groups]"

303

​

304

# -- Matcher function for Casbin, `glob` for glob matcher and `regex` for regex matcher.

305

policy.matchMode: "glob"

306

​

307

# GnuPG public keys for commit verification

308

## Ref: https://argo-cd.readthedocs.io/en/stable/user-guide/gpg-verification/

309

gpg:

310

# -- Annotations to be added to argocd-gpg-keys-cm configmap

311

annotations: {}

312

​

313

# -- [GnuPG] public keys to add to the keyring

314

# @default -- `{}` (See [values.yaml])

315

## Note: Public keys should be exported with `gpg --export --armor <KEY>`

316

keys: {}

317

# 4AEE18F83AFDEB23: |

318

# -----BEGIN PGP PUBLIC KEY BLOCK-----

319

# ...

320

# -----END PGP PUBLIC KEY BLOCK-----

321

​

322

# SSH known hosts for Git repositories

323

## Ref: https://argo-cd.readthedocs.io/en/stable/operator-manual/declarative-setup/#ssh-known-host-public-keys

324

ssh:

325

# -- Annotations to be added to argocd-ssh-known-hosts-cm configmap

326

annotations: {}

327

​

328

# -- Known hosts to be added to the known host list by default.

329

# @default -- See [values.yaml]

330

knownHosts: |

331

[ssh.github.com]:443 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBEmKSENjQEezOmxkZMy7opKgwFB9nkt5YRrYMjNuG5N87uRgg6CLrbo5wAdT/y6v0mKV0U2w0WZ2YB/++Tpockg=

332

[ssh.github.com]:443 ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOMqqnkVzrm0SdG6UOoqKLsabgH5C9okWi0dh2l9GKJl

333

[ssh.github.com]:443 ssh-rsa 

334

bitbucket.org ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBPIQmuzMBuKdWeF4+a2sjSSpBK0iqitSQ+5BM9KhpexuGt20JpTVM7u5BDZngncgrqDMbWdxMWWOGtZ9UgbqgZE=

335

bitbucket.org ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIIazEu89wgQZ4bqs3d63QSMzYVa0MuJ2e2gKTKqu+UUO

336

bitbucket.org ssh-rsa 

337

github.com ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBEmKSENjQEezOmxkZMy7opKgwFB9nkt5YRrYMjNuG5N87uRgg6CLrbo5wAdT/y6v0mKV0U2w0WZ2YB/++Tpockg=

338

github.com ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIOMqqnkVzrm0SdG6UOoqKLsabgH5C9okWi0dh2l9GKJl

339

github.com ssh-rsa 

340

gitlab.com ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBFSMqzJeV9rUzU4kWitGjeR4PWSa29SPqJ1fVkhtj3Hw9xjLVXVYrU9QlYWrOLXBpQ6KWjbjTDTdDkoohFzgbEY=

341

gitlab.com ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIAfuCHKVTjquxvt6CM6tdG4SLp1Btn/nOeHHE5UOzRdf

342

gitlab.com ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCsj2bNKTBSpIYDEGk9KxsGh3mySTRgMtXL583qmBpzeQ+jqCMRgBqB98u3z++J1sKlXHWfM9dyhSevkMwSbhoR8XIq/U0tCNyokEi/ueaBMCvbcTHhO7FcwzY92WK4Yt0aGROY5qX2UKSeOvuP4D6TPqKF1onrSzH9bx9XUf2lEdWT/ia1NEKjunUqu1xOB/StKDHMoX4/OKyIzuS0q/T1zOATthvasJFoPrAjkohTyaDUz2LN5JoH839hViyEG82yB+MjcFV5MU3N1l1QL3cVUCh93xSaua1N85qivl+siMkPGbO5xR/En4iEY6K2XPASUEMaieWVNTRCtJ4S8H+9

343

ssh.dev.azure.com ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC7Hr1oTWqNqOlzGJOfGJ4NakVyIzf1rXYd4d7wo6jBlkLvCA4odBlL0mDUyZ0/QUfTTqeu+tm22gOsv+VrVTMk6vwRU75gY/y9ut5Mb3bR5BV58dKXyq9A9UeB5Cakehn5Zgm6x1mKoVyf+FFn26iYqXJRgzIZZcZ5V6hrE0Qg39kZm4az48o0AUbf6Sp4SLdvnuMa2sVNwHBboS7EJkm57XQPVU3/QpyNLHbWDdzwtrlS+ez30S3AdYhLKEOxAG8weOnyrtLJAUen9mTkol8oII1edf7mWWbWVf0nBmly21+nZcmCTISQBtdcyPaEno7fFQMDD26/s0lfKob4Kw8H

344

vs-ssh.visualstudio.com ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC7Hr1oTWqNqOlzGJOfGJ4NakVyIzf1rXYd4d7wo6jBlkLvCA4odBlL0mDUyZ0/QUfTTqeu+tm22gOsv+VrVTMk6vwRU75gY/y9ut5Mb3bR5BV58dKXyq9A9UeB5Cakehn5Zgm6x1mKoVyf+FFn26iYqXJRgzIZZcZ5V6hrE0Qg39kZm4az48o0AUbf6Sp4SLdvnuMa2sVNwHBboS7EJkm57XQPVU3/QpyNLHbWDdzwtrlS+ez30S3AdYhLKEOxAG8weOnyrtLJAUen9mTkol8oII1edf7mWWbWVf0nBmly21+nZcmCTISQBtdcyPaEno7fFQMDD26/s0lfKob4Kw8H

345

​

346

# -- Additional known hosts for private repositories

347

extraHosts: ''

348

​

349

# Repository TLS certificates

350

# Ref: https://argo-cd.readthedocs.io/en/stable/operator-manual/declarative-setup/#repositories-using-self-signed-tls-certificates-or-are-signed-by-custom-ca

351

tls:

352

# -- Annotations to be added to argocd-tls-certs-cm configmap

353

annotations: {}

354

​

355

# -- TLS certificates for Git repositories

356

# @default -- `{}` (See [values.yaml])

357

certificates: {}

358

# server.example.com: |

359

# -----BEGIN CERTIFICATE-----

360

# ...

361

# -----END CERTIFICATE-----

362

​

363

# ConfigMap for Config Management Plugins

364

# Ref: https://argo-cd.readthedocs.io/en/stable/operator-manual/config-management-plugins/

365

cmp:

366

# -- Create the argocd-cmp-cm configmap

367

create: false

368

​

369

# -- Annotations to be added to argocd-cmp-cm configmap

370

annotations: {}

371

​

372

# -- Plugin yaml files to be added to argocd-cmp-cm

373

plugins: {}

374

# --- First plugin

375

# my-plugin:

376

# init:

377

# command: [sh]

378

# args: [-c, 'echo "Initializing..."']

379

# generate:

380

# command: [sh, -c]

381

# args:

382

# - |

383

# echo "{\"kind\": \"ConfigMap\", \"apiVersion\": \"v1\", \"metadata\": { \"name\": \"$ARGOCD\_APP\_NAME\", \"namespace\": \"$ARGOCD\_APP\_NAMESPACE\", \"annotations\": {\"Foo\": \"$ARGOCD\_ENV\_FOO\", \"KubeVersion\": \"$KUBE\_VERSION\", \"KubeApiVersion\": \"$KUBE\_API\_VERSIONS\",\"Bar\": \"baz\"}}}"

384

# discover:

385

# fileName: "./subdir/s\*.yaml"

386

# find:

387

# glob: "\*\*/Chart.yaml"

388

# command: [sh, -c, find . -name env.yaml]

389

​

390

# --- Second plugin

391

# my-plugin2:

392

# init:

393

# command: [sh]

394

# args: [-c, 'echo "Initializing..."']

395

# generate:

396

# command: [sh, -c]

397

# args:

398

# - |

399

# echo "{\"kind\": \"ConfigMap\", \"apiVersion\": \"v1\", \"metadata\": { \"name\": \"$ARGOCD\_APP\_NAME\", \"namespace\": \"$ARGOCD\_APP\_NAMESPACE\", \"annotations\": {\"Foo\": \"$ARGOCD\_ENV\_FOO\", \"KubeVersion\": \"$KUBE\_VERSION\", \"KubeApiVersion\": \"$KUBE\_API\_VERSIONS\",\"Bar\": \"baz\"}}}"

400

# discover:

401

# fileName: "./subdir/s\*.yaml"

402

# find:

403

# glob: "\*\*/Chart.yaml"

404

# command: [sh, -c, find . -name env.yaml]

405

​

406

# -- Provide one or multiple [external cluster credentials]

407

# @default -- `[]` (See [values.yaml])

408

## Ref:

409

## - https://argo-cd.readthedocs.io/en/stable/operator-manual/declarative-setup/#clusters

410

## - https://argo-cd.readthedocs.io/en/stable/operator-manual/security/#external-cluster-credentials

411

## - https://argo-cd.readthedocs.io/en/stable/user-guide/projects/#project-scoped-repositories-and-clusters

412

clusterCredentials: []

413

# - name: mycluster

414

# server: https://mycluster.example.com

415

# labels: {}

416

# annotations: {}

417

# config:

418

# bearerToken: "<authentication token>"

419

# tlsClientConfig:

420

# insecure: false

421

# caData: "<base64 encoded certificate>"

422

# - name: mycluster2

423

# server: https://mycluster2.example.com

424

# labels: {}

425

# annotations: {}

426

# namespaces: namespace1,namespace2

427

# clusterResources: true

428

# config:

429

# bearerToken: "<authentication token>"

430

# tlsClientConfig:

431

# insecure: false

432

# caData: "<base64 encoded certificate>"

433

# - name: mycluster3-project-scoped

434

# server: https://mycluster3.example.com

435

# labels: {}

436

# annotations: {}

437

# project: my-project1

438

# config:

439

# bearerToken: "<authentication token>"

440

# tlsClientConfig:

441

# insecure: false

442

# caData: "<base64 encoded certificate>"

443

​

444

# -- Repository credentials to be used as Templates for other repos

445

## Creates a secret for each key/value specified below to create repository credentials

446

credentialTemplates: {}

447

# github-enterprise-creds-1:

448

# url: https://github.com/argoproj

449

# githubAppID: 1

450

# githubAppInstallationID: 2

451

# githubAppEnterpriseBaseUrl: https://ghe.example.com/api/v3

452

# githubAppPrivateKey: |

453

# -----BEGIN OPENSSH PRIVATE KEY-----

454

# ...

455

# -----END OPENSSH PRIVATE KEY-----

456

# https-creds:

457

# url: https://github.com/argoproj

458

# password: my-password

459

# username: my-username

460

# ssh-creds:

461

# url: [email protected]:argoproj-labs

462

# sshPrivateKey: |

463

# -----BEGIN OPENSSH PRIVATE KEY-----

464

# ...

465

# -----END OPENSSH PRIVATE KEY-----

466

​

467

# -- Annotations to be added to `configs.credentialTemplates` Secret

468

credentialTemplatesAnnotations: {}

469

​

470

# -- Repositories list to be used by applications

471

## Creates a secret for each key/value specified below to create repositories

472

## Note: the last example in the list would use a repository credential template, configured under "configs.credentialTemplates".

473

repositories: {}

474

# istio-helm-repo:

475

# url: https://storage.googleapis.com/istio-prerelease/daily-build/master-latest-daily/charts

476

# name: istio.io

477

# type: helm

478

# private-helm-repo:

479

# url: https://my-private-chart-repo.internal

480

# name: private-repo

481

# type: helm

482

# password: my-password

483

# username: my-username

484

# private-repo:

485

# url: https://github.com/argoproj/private-repo

486

​

487

# -- Annotations to be added to `configs.repositories` Secret

488

repositoriesAnnotations: {}

489

​

490

# Argo CD sensitive data

491

# Ref: https://argo-cd.readthedocs.io/en/stable/operator-manual/user-management/#sensitive-data-and-sso-client-secrets

492

secret:

493

# -- Create the argocd-secret

494

createSecret: true

495

# -- Labels to be added to argocd-secret

496

labels: {}

497

# -- Annotations to be added to argocd-secret

498

annotations: {}

499

​

500

# -- Shared secret for authenticating GitHub webhook events

501

githubSecret: ""

502

# -- Shared secret for authenticating GitLab webhook events

503

gitlabSecret: ""

504

# -- Shared secret for authenticating BitbucketServer webhook events

505

bitbucketServerSecret: ""

506

# -- UUID for authenticating Bitbucket webhook events

507

bitbucketUUID: ""

508

# -- Shared secret for authenticating Gogs webhook events

509

gogsSecret: ""

510

## Azure DevOps

511

azureDevops:

512

# -- Shared secret username for authenticating Azure DevOps webhook events

513

username: ""

514

# -- Shared secret password for authenticating Azure DevOps webhook events

515

password: ""

516

​

517

# -- add additional secrets to be added to argocd-secret

518

## Custom secrets. Useful for injecting SSO secrets into environment variables.

519

## Ref: https://argo-cd.readthedocs.io/en/stable/operator-manual/user-management/#sensitive-data-and-sso-client-secrets

520

## Note that all values must be non-empty.

521

extra:

522

{}

523

# LDAP\_PASSWORD: "mypassword"

524

​

525

# -- Bcrypt hashed admin password

526

## Argo expects the password in the secret to be bcrypt hashed. You can create this hash with

527

## `htpasswd -nbBC 10 "" $ARGO\_PWD | tr -d ':\n' | sed 's/$2y/$2a/'`

528

argocdServerAdminPassword: ""

529

# -- Admin password modification time. Eg. `"2006-01-02T15:04:05Z"`

530

# @default -- `""` (defaults to current time)

531

argocdServerAdminPasswordMtime: ""

532

​

533

# -- Define custom [CSS styles] for your argo instance.

534

# This setting will automatically mount the provided CSS and reference it in the argo configuration.

535

# @default -- `""` (See [values.yaml])

536

## Ref: https://argo-cd.readthedocs.io/en/stable/operator-manual/custom-styles/

537

styles: ""

538

# styles: |

539

# .sidebar {

540

# background: linear-gradient(to bottom, #999, #777, #333, #222, #111);

541

# }

542

​

543

# -- Array of extra K8s manifests to deploy

544

## Note: Supports use of custom Helm templates

545

extraObjects: []

546

# - apiVersion: secrets-store.csi.x-k8s.io/v1

547

# kind: SecretProviderClass

548

# metadata:

549

# name: argocd-secrets-store

550

# spec:

551

# provider: aws

552

# parameters:

553

# objects: |

554

# - objectName: "argocd"

555

# objectType: "secretsmanager"

556

# jmesPath:

557

# - path: "client\_id"

558

# objectAlias: "client\_id"

559

# - path: "client\_secret"

560

# objectAlias: "client\_secret"

561

# secretObjects:

562

# - data:

563

# - key: client\_id

564

# objectName: client\_id

565

# - key: client\_secret

566

# objectName: client\_secret

567

# secretName: argocd-secrets-store

568

# type: Opaque

569

# labels:

570

# app.kubernetes.io/part-of: argocd

571

​

572

## Application controller

573

controller:

574

# -- Application controller name string

575

name: application-controller

576

​

577

# -- The number of application controller pods to run.

578

# Additional replicas will cause sharding of managed clusters across number of replicas.

579

replicas: 1

580

​

581

# -- Maximum number of controller revisions that will be maintained in StatefulSet history

582

revisionHistoryLimit: 5

583

​

584

## Application controller Pod Disruption Budget

585

## Ref: https://kubernetes.io/docs/tasks/run-application/configure-pdb/

586

pdb:

587

# -- Deploy a [PodDisruptionBudget] for the application controller

588

enabled: false

589

# -- Labels to be added to application controller pdb

590

labels: {}

591

# -- Annotations to be added to application controller pdb

592

annotations: {}

593

# -- Number of pods that are available after eviction as number or percentage (eg.: 50%)

594

# @default -- `""` (defaults to 0 if not specified)

595

minAvailable: ""

596

# -- Number of pods that are unavailable after eviction as number or percentage (eg.: 50%).

597

## Has higher precedence over `controller.pdb.minAvailable`

598

maxUnavailable: ""

599

​

600

## Application controller image

601

image:

602

# -- Repository to use for the application controller

603

# @default -- `""` (defaults to global.image.repository)

604

repository: ""

605

# -- Tag to use for the application controller

606

# @default -- `""` (defaults to global.image.tag)

607

tag: ""

608

# -- Image pull policy for the application controller

609

# @default -- `""` (defaults to global.image.imagePullPolicy)

610

imagePullPolicy: ""

611

​

612

# -- Secrets with credentials to pull images from a private registry

613

# @default -- `[]` (defaults to global.imagePullSecrets)

614

imagePullSecrets: []

615

​

616

# -- Additional command line arguments to pass to application controller

617

extraArgs: []

618

​

619

# -- Environment variables to pass to application controller

620

env: []

621

​

622

# -- envFrom to pass to application controller

623

# @default -- `[]` (See [values.yaml])

624

envFrom: []

625

# - configMapRef:

626

# name: config-map-name

627

# - secretRef:

628

# name: secret-name

629

​

630

# -- Additional containers to be added to the application controller pod

631

## Note: Supports use of custom Helm templates

632

extraContainers: []

633

​

634

# -- Init containers to add to the application controller pod

635

## If your target Kubernetes cluster(s) require a custom credential (exec) plugin

636

## you could use this (and the same in the server pod) to provide such executable

637

## Ref: https://kubernetes.io/docs/reference/access-authn-authz/authentication/#client-go-credential-plugins

638

## Note: Supports use of custom Helm templates

639

initContainers: []

640

# - name: download-tools

641

# image: alpine:3

642

# command: [sh, -c]

643

# args:

644

# - wget -qO kubelogin.zip https://github.com/Azure/kubelogin/releases/download/v0.0.25/kubelogin-linux-amd64.zip &&

645

# unzip kubelogin.zip && mv bin/linux\_amd64/kubelogin /custom-tools/

646

# volumeMounts:

647

# - mountPath: /custom-tools

648

# name: custom-tools

649

​

650

# -- Additional volumeMounts to the application controller main container

651

volumeMounts: []

652

# - mountPath: /usr/local/bin/kubelogin

653

# name: custom-tools

654

# subPath: kubelogin

655

​

656

# -- Additional volumes to the application controller pod

657

volumes: []

658

# - name: custom-tools

659

# emptyDir: {}

660

​

661

# -- Annotations for the application controller StatefulSet

662

statefulsetAnnotations: {}

663

​

664

# -- Annotations to be added to application controller pods

665

podAnnotations: {}

666

​

667

# -- Labels to be added to application controller pods

668

podLabels: {}

669

​

670

# -- Resource limits and requests for the application controller pods

671

resources: {}

672

# limits:

673

# cpu: 500m

674

# memory: 512Mi

675

# requests:

676

# cpu: 250m

677

# memory: 256Mi

678

​

679

# Application controller container ports

680

containerPorts:

681

# -- Metrics container port

682

metrics: 8082

683

​

684

# -- Host Network for application controller pods

685

hostNetwork: false

686

​

687

# -- [DNS configuration]

688

dnsConfig: {}

689

# -- Alternative DNS policy for application controller pods

690

dnsPolicy: "ClusterFirst"

691

​

692

# -- Application controller container-level security context

693

# @default -- See [values.yaml]

694

containerSecurityContext:

695

runAsNonRoot: true

696

readOnlyRootFilesystem: true

697

allowPrivilegeEscalation: false

698

seccompProfile:

699

type: RuntimeDefault

700

capabilities:

701

drop:

702

- ALL

703

​

704

# Readiness probe for application controller

705

## Ref: https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-probes/

706

readinessProbe:

707

# -- Minimum consecutive failures for the [probe] to be considered failed after having succeeded

708

failureThreshold: 3

709

# -- Number of seconds after the container has started before [probe] is initiated

710

initialDelaySeconds: 10

711

# -- How often (in seconds) to perform the [probe]

712

periodSeconds: 10

713

# -- Minimum consecutive successes for the [probe] to be considered successful after having failed

714

successThreshold: 1

715

# -- Number of seconds after which the [probe] times out

716

timeoutSeconds: 1

717

​

718

# -- terminationGracePeriodSeconds for container lifecycle hook

719

terminationGracePeriodSeconds: 30

720

​

721

# -- Priority class for the application controller pods

722

# @default -- `""` (defaults to global.priorityClassName)

723

priorityClassName: ""

724

​

725

# -- [Node selector]

726

# @default -- `{}` (defaults to global.nodeSelector)

727

nodeSelector: {}

728

​

729

# -- [Tolerations] for use with node taints

730

# @default -- `[]` (defaults to global.tolerations)

731

tolerations: []

732

​

733

# -- Assign custom [affinity] rules to the deployment

734

# @default -- `{}` (defaults to global.affinity preset)

735

affinity: {}

736

​

737

# -- Assign custom [TopologySpreadConstraints] rules to the application controller

738

# @default -- `[]` (defaults to global.topologySpreadConstraints)

739

## Ref: https://kubernetes.io/docs/concepts/workloads/pods/pod-topology-spread-constraints/

740

## If labelSelector is left out, it will default to the labelSelector configuration of the deployment

741

topologySpreadConstraints: []

742

# - maxSkew: 1

743

# topologyKey: topology.kubernetes.io/zone

744

# whenUnsatisfiable: DoNotSchedule

745

​

746

serviceAccount:

747

# -- Create a service account for the application controller

748

create: true

749

# -- Service account name

750

name: argocd-application-controller

751

# -- Annotations applied to created service account

752

annotations: {}

753

# -- Labels applied to created service account

754

labels: {}

755

# -- Automount API credentials for the Service Account

756

automountServiceAccountToken: true

757

​

758

## Application controller metrics configuration

759

metrics:

760

# -- Deploy metrics service

761

enabled: false

762

# -- Prometheus ServiceMonitor scrapeTimeout. If empty, Prometheus uses the global scrape timeout unless it is less than the target's scrape interval value in which the latter is used.

763

scrapeTimeout: ""

764

applicationLabels:

765

# -- Enables additional labels in argocd\_app\_labels metric

766

enabled: false

767

# -- Additional labels

768

labels: []

769

service:

770

# -- Metrics service type

771

type: ClusterIP

772

# -- Metrics service clusterIP. `None` makes a "headless service" (no virtual IP)

773

clusterIP: ""

774

# -- Metrics service annotations

775

annotations: {}

776

# -- Metrics service labels

777

labels: {}

778

# -- Metrics service port

779

servicePort: 8082

780

# -- Metrics service port name

781

portName: http-metrics

782

serviceMonitor:

783

# -- Enable a prometheus ServiceMonitor

784

enabled: false

785

# -- Prometheus ServiceMonitor interval

786

interval: 30s

787

# -- Prometheus [RelabelConfigs] to apply to samples before scraping

788

relabelings: []

789

# -- Prometheus [MetricRelabelConfigs] to apply to samples before ingestion

790

metricRelabelings: []

791

# -- Prometheus ServiceMonitor selector

792

selector: {}

793

# prometheus: kube-prometheus

794

​

795

# -- Prometheus ServiceMonitor scheme

796

scheme: ""

797

# -- Prometheus ServiceMonitor tlsConfig

798

tlsConfig: {}

799

# -- Prometheus ServiceMonitor namespace

800

namespace: "" # "monitoring"

801

# -- Prometheus ServiceMonitor labels

802

additionalLabels: {}

803

# -- Prometheus ServiceMonitor annotations

804

annotations: {}

805

rules:

806

# -- Deploy a PrometheusRule for the application controller

807

enabled: false

808

# -- PrometheusRule namespace

809

namespace: "" # "monitoring"

810

# -- PrometheusRule selector

811

selector: {}

812

# prometheus: kube-prometheus

813

​

814

# -- PrometheusRule labels

815

additionalLabels: {}

816

# -- PrometheusRule annotations

817

annotations: {}

818

​

819

# -- PrometheusRule.Spec for the application controller

820

spec: []

821

# - alert: ArgoAppMissing

822

# expr: |

823

# absent(argocd\_app\_info) == 1

824

# for: 15m

825

# labels:

826

# severity: critical

827

# annotations:

828

# summary: "[Argo CD] No reported applications"

829

# description: >

830

# Argo CD has not reported any applications data for the past 15 minutes which

831

# means that it must be down or not functioning properly. This needs to be

832

# resolved for this cloud to continue to maintain state.

833

# - alert: ArgoAppNotSynced

834

# expr: |

835

# argocd\_app\_info{sync\_status!="Synced"} == 1

836

# for: 12h

837

# labels:

838

# severity: warning

839

# annotations:

840

# summary: "[{{`{{$labels.name}}`}}] Application not synchronized"

841

# description: >

842

# The application [{{`{{$labels.name}}`}} has not been synchronized for over

843

# 12 hours which means that the state of this cloud has drifted away from the

844

# state inside Git.

845

​

846

## Enable this and set the rules: to whatever custom rules you want for the Cluster Role resource.

847

## Defaults to off

848

clusterRoleRules:

849

# -- Enable custom rules for the application controller's ClusterRole resource

850

enabled: false

851

# -- List of custom rules for the application controller's ClusterRole resource

852

rules: []

853

​

854

## Dex

855

dex:

856

# -- Enable dex

857

enabled: true

858

# -- Dex name

859

name: dex-server

860

​

861

# -- Additional command line arguments to pass to the Dex server

862

extraArgs: []

863

​

864

metrics:

865

# -- Deploy metrics service

866

enabled: false

867

service:

868

# -- Metrics service annotations

869

annotations: {}

870

# -- Metrics service labels

871

labels: {}

872

# -- Metrics service port name

873

portName: http-metrics

874

serviceMonitor:

875

# -- Enable a prometheus ServiceMonitor

876

enabled: false

877

# -- Prometheus ServiceMonitor interval

878

interval: 30s

879

# -- Prometheus [RelabelConfigs] to apply to samples before scraping

880

relabelings: []

881

# -- Prometheus [MetricRelabelConfigs] to apply to samples before ingestion

882

metricRelabelings: []

883

# -- Prometheus ServiceMonitor selector

884

selector: {}

885

# prometheus: kube-prometheus

886

​

887

# -- Prometheus ServiceMonitor scheme

888

scheme: ""

889

# -- Prometheus ServiceMonitor tlsConfig

890

tlsConfig: {}

891

# -- Prometheus ServiceMonitor namespace

892

namespace: "" # "monitoring"

893

# -- Prometheus ServiceMonitor labels

894

additionalLabels: {}

895

# -- Prometheus ServiceMonitor annotations

896

annotations: {}

897

​

898

## Dex Pod Disruption Budget

899

## Ref: https://kubernetes.io/docs/tasks/run-application/configure-pdb/

900

pdb:

901

# -- Deploy a [PodDisruptionBudget] for the Dex server

902

enabled: false

903

# -- Labels to be added to Dex server pdb

904

labels: {}

905

# -- Annotations to be added to Dex server pdb

906

annotations: {}

907

# -- Number of pods that are available after eviction as number or percentage (eg.: 50%)

908

# @default -- `""` (defaults to 0 if not specified)

909

minAvailable: ""

910

# -- Number of pods that are unavailble after eviction as number or percentage (eg.: 50%).

911

## Has higher precedence over `dex.pdb.minAvailable`

912

maxUnavailable: ""

913

​

914

## Dex image

915

image:

916

# -- Dex image repository

917

repository: ghcr.io/dexidp/dex

918

# -- Dex image tag

919

tag: v2.38.0

920

# -- Dex imagePullPolicy

921

# @default -- `""` (defaults to global.image.imagePullPolicy)

922

imagePullPolicy: ""

923

​

924

# -- Secrets with credentials to pull images from a private registry

925

# @default -- `[]` (defaults to global.imagePullSecrets)

926

imagePullSecrets: []

927

​

928

# Argo CD init image that creates Dex config

929

initImage:

930

# -- Argo CD init image repository

931

# @default -- `""` (defaults to global.image.repository)

932

repository: ""

933

# -- Argo CD init image tag

934

# @default -- `""` (defaults to global.image.tag)

935

tag: ""

936

# -- Argo CD init image imagePullPolicy

937

# @default -- `""` (defaults to global.image.imagePullPolicy)

938

imagePullPolicy: ""

939

# -- Argo CD init image resources

940

# @default -- `{}` (defaults to dex.resources)

941

resources: {}

942

# requests:

943

# cpu: 5m

944

# memory: 96Mi

945

# limits:

946

# cpu: 10m

947

# memory: 144Mi

948

​

949

# -- Environment variables to pass to the Dex server

950

env: []

951

​

952

# -- envFrom to pass to the Dex server

953

# @default -- `[]` (See [values.yaml])

954

envFrom: []

955

# - configMapRef:

956

# name: config-map-name

957

# - secretRef:

958

# name: secret-name

959

​

960

# -- Additional containers to be added to the dex pod

961

## Note: Supports use of custom Helm templates

962

extraContainers: []

963

​

964

# -- Init containers to add to the dex pod

965

## Note: Supports use of custom Helm templates

966

initContainers: []

967

​

968

# -- Additional volumeMounts to the dex main container

969

volumeMounts: []

970

​

971

# -- Additional volumes to the dex pod

972

volumes: []

973

​

974

# TLS certificate configuration via Secret

975

## Ref: https://argo-cd.readthedocs.io/en/stable/operator-manual/tls/#configuring-tls-to-argocd-dex-server

976

## Note: Issuing certificates via cert-manager in not supported right now because it's not possible to restart Dex automatically without extra controllers.

977

certificateSecret:

978

# -- Create argocd-dex-server-tls secret

979

enabled: false

980

# -- Labels to be added to argocd-dex-server-tls secret

981

labels: {}

982

# -- Annotations to be added to argocd-dex-server-tls secret

983

annotations: {}

984

# -- Certificate authority. Required for self-signed certificates.

985

ca: ''

986

# -- Certificate private key

987

key: ''

988

# -- Certificate data. Must contain SANs of Dex service (ie: argocd-dex-server, argocd-dex-server.argo-cd.svc)

989

crt: ''

990

​

991

# -- Annotations to be added to the Dex server Deployment

992

deploymentAnnotations: {}

993

​

994

# -- Annotations to be added to the Dex server pods

995

podAnnotations: {}

996

​

997

# -- Labels to be added to the Dex server pods

998

podLabels: {}

999

​

1000

# -- Resource limits and requests for dex

1001

resources: {}

1002

# limits:

1003

# cpu: 50m

1004

# memory: 64Mi

1005

# requests:

1006

# cpu: 10m

1007

# memory: 32Mi

1008

​

1009

# Dex container ports

1010

# NOTE: These ports are currently hardcoded and cannot be changed

1011

containerPorts:

1012

# -- HTTP container port

1013

http: 5556

1014

# -- gRPC container port

1015

grpc: 5557

1016

# -- Metrics container port

1017

metrics: 5558

1018

​

1019

# -- [DNS configuration]

1020

dnsConfig: {}

1021

# -- Alternative DNS policy for Dex server pods

1022

dnsPolicy: "ClusterFirst"

1023

​

1024

# -- Dex container-level security context

1025

# @default -- See [values.yaml]

1026

containerSecurityContext:

1027

runAsNonRoot: true

1028

readOnlyRootFilesystem: true

1029

allowPrivilegeEscalation: false

1030

seccompProfile:

1031

type: RuntimeDefault

1032

capabilities:

1033

drop:

1034

- ALL

1035

​

1036

## Probes for Dex server

1037

## Supported from Dex >= 2.28.0

1038

livenessProbe:

1039

# -- Enable Kubernetes liveness probe for Dex >= 2.28.0

1040

enabled: false

1041

# -- Minimum consecutive failures for the [probe] to be considered failed after having succeeded

1042

failureThreshold: 3

1043

# -- Number of seconds after the container has started before [probe] is initiated

1044

initialDelaySeconds: 10

1045

# -- How often (in seconds) to perform the [probe]

1046

periodSeconds: 10

1047

# -- Minimum consecutive successes for the [probe] to be considered successful after having failed

1048

successThreshold: 1

1049

# -- Number of seconds after which the [probe] times out

1050

timeoutSeconds: 1

1051

​

1052

readinessProbe:

1053

# -- Enable Kubernetes readiness probe for Dex >= 2.28.0

1054

enabled: false

1055

# -- Minimum consecutive failures for the [probe] to be considered failed after having succeeded

1056

failureThreshold: 3

1057

# -- Number of seconds after the container has started before [probe] is initiated

1058

initialDelaySeconds: 10

1059

# -- How often (in seconds) to perform the [probe]

1060

periodSeconds: 10

1061

# -- Minimum consecutive successes for the [probe] to be considered successful after having failed

1062

successThreshold: 1

1063

# -- Number of seconds after which the [probe] times out

1064

timeoutSeconds: 1

1065

​

1066

# -- terminationGracePeriodSeconds for container lifecycle hook

1067

terminationGracePeriodSeconds: 30

1068

​

1069

serviceAccount:

1070

# -- Create dex service account

1071

create: true

1072

# -- Dex service account name

1073

name: argocd-dex-server

1074

# -- Annotations applied to created service account

1075

annotations: {}

1076

# -- Automount API credentials for the Service Account

1077

automountServiceAccountToken: true

1078

​

1079

# -- Service port for HTTP access

1080

servicePortHttp: 5556

1081

# -- Service port name for HTTP access

1082

servicePortHttpName: http

1083

# -- Service port for gRPC access

1084

servicePortGrpc: 5557

1085

# -- Service port name for gRPC access

1086

servicePortGrpcName: grpc

1087

# -- Service port for metrics access

1088

servicePortMetrics: 5558

1089

​

1090

# -- Priority class for the dex pods

1091

# @default -- `""` (defaults to global.priorityClassName)

1092

priorityClassName: ""

1093

​

1094

# -- [Node selector]

1095

# @default -- `{}` (defaults to global.nodeSelector)

1096

nodeSelector: {}

1097

​

1098

# -- [Tolerations] for use with node taints

1099

# @default -- `[]` (defaults to global.tolerations)

1100

tolerations: []

1101

​

1102

# -- Assign custom [affinity] rules to the deployment

1103

# @default -- `{}` (defaults to global.affinity preset)

1104

affinity: {}

1105

​

1106

# -- Assign custom [TopologySpreadConstraints] rules to dex

1107

# @default -- `[]` (defaults to global.topologySpreadConstraints)

1108

## Ref: https://kubernetes.io/docs/concepts/workloads/pods/pod-topology-spread-constraints/

1109

## If labelSelector is left out, it will default to the labelSelector configuration of the deployment

1110

topologySpreadConstraints: []

1111

# - maxSkew: 1

1112

# topologyKey: topology.kubernetes.io/zone

1113

# whenUnsatisfiable: DoNotSchedule

1114

​

1115

# -- Deployment strategy to be added to the Dex server Deployment

1116

deploymentStrategy: {}

1117

# type: RollingUpdate

1118

# rollingUpdate:

1119

# maxSurge: 25%

1120

# maxUnavailable: 25%

1121

​

1122

# -- Dex log format. Either `text` or `json`

1123

# @default -- `""` (defaults to global.logging.format)

1124

logFormat: ""

1125

# -- Dex log level. One of: `debug`, `info`, `warn`, `error`

1126

# @default -- `""` (defaults to global.logging.level)

1127

logLevel: ""

1128

​

1129

## Redis

1130

redis:

1131

# -- Enable redis

1132

enabled: true

1133

# -- Redis name

1134

name: redis

1135

​

1136

## Redis Pod Disruption Budget

1137

## Ref: https://kubernetes.io/docs/tasks/run-application/configure-pdb/

1138

pdb:

1139

# -- Deploy a [PodDisruptionBudget] for the Redis

1140

enabled: false

1141

# -- Labels to be added to Redis pdb

1142

labels: {}

1143

# -- Annotations to be added to Redis pdb

1144

annotations: {}

1145

# -- Number of pods that are available after eviction as number or percentage (eg.: 50%)

1146

# @default -- `""` (defaults to 0 if not specified)

1147

minAvailable: ""

1148

# -- Number of pods that are unavailble after eviction as number or percentage (eg.: 50%).

1149

## Has higher precedence over `redis.pdb.minAvailable`

1150

maxUnavailable: ""

1151

​

1152

## Redis image

1153

image:

1154

# -- Redis repository

1155

repository: public.ecr.aws/docker/library/redis

1156

# -- Redis tag

1157

tag: 7.0.15-alpine

1158

# -- Redis image pull policy

1159

# @default -- `""` (defaults to global.image.imagePullPolicy)

1160

imagePullPolicy: ""

1161

​

1162

## Prometheus redis-exporter sidecar

1163

exporter:

1164

# -- Enable Prometheus redis-exporter sidecar

1165

enabled: false

1166

# -- Environment variables to pass to the Redis exporter

1167

env: []

1168

## Prometheus redis-exporter image

1169

image:

1170

# -- Repository to use for the redis-exporter

1171

repository: public.ecr.aws/bitnami/redis-exporter

1172

# -- Tag to use for the redis-exporter

1173

tag: 1.57.0

1174

# -- Image pull policy for the redis-exporter

1175

# @default -- `""` (defaults to global.image.imagePullPolicy)

1176

imagePullPolicy: ""

1177

​

1178

# -- Redis exporter security context

1179

# @default -- See [values.yaml]

1180

containerSecurityContext:

1181

runAsNonRoot: true

1182

readOnlyRootFilesystem: true

1183

allowPrivilegeEscalation: false

1184

seccompProfile:

1185

type: RuntimeDefault

1186

capabilities:

1187

drop:

1188

- ALL

1189

​

1190

## Probes for Redis exporter (optional)

1191

## Ref: https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-probes/

1192

readinessProbe:

1193

# -- Enable Kubernetes liveness probe for Redis exporter (optional)

1194

enabled: false

1195

# -- Number of seconds after the container has started before [probe] is initiated

1196

initialDelaySeconds: 30

1197

# -- How often (in seconds) to perform the [probe]

1198

periodSeconds: 15

1199

# -- Number of seconds after which the [probe] times out

1200

timeoutSeconds: 15

1201

# -- Minimum consecutive successes for the [probe] to be considered successful after having failed

1202

successThreshold: 1

1203

# -- Minimum consecutive failures for the [probe] to be considered failed after having succeeded

1204

failureThreshold: 5

1205

livenessProbe:

1206

# -- Enable Kubernetes liveness probe for Redis exporter

1207

enabled: false

1208

# -- Number of seconds after the container has started before [probe] is initiated

1209

initialDelaySeconds: 30

1210

# -- How often (in seconds) to perform the [probe]

1211

periodSeconds: 15

1212

# -- Number of seconds after which the [probe] times out

1213

timeoutSeconds: 15

1214

# -- Minimum consecutive successes for the [probe] to be considered successful after having failed

1215

successThreshold: 1

1216

# -- Minimum consecutive failures for the [probe] to be considered failed after having succeeded

1217

failureThreshold: 5

1218

​

1219

# -- Resource limits and requests for redis-exporter sidecar

1220

resources: {}

1221

# limits:

1222

# cpu: 50m

1223

# memory: 64Mi

1224

# requests:

1225

# cpu: 10m

1226

# memory: 32Mi

1227

​

1228

# -- Secrets with credentials to pull images from a private registry

1229

# @default -- `[]` (defaults to global.imagePullSecrets)

1230

imagePullSecrets: []

1231

​

1232

# -- Additional command line arguments to pass to redis-server

1233

extraArgs: []

1234

# - --bind

1235

# - "0.0.0.0"

1236

​

1237

# -- Environment variables to pass to the Redis server

1238

env: []

1239

​

1240

# -- envFrom to pass to the Redis server

1241

# @default -- `[]` (See [values.yaml])

1242

envFrom: []

1243

# - configMapRef:

1244

# name: config-map-name

1245

# - secretRef:

1246

# name: secret-name

1247

​

1248

## Probes for Redis server (optional)

1249

## Ref: https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-probes/

1250

readinessProbe:

1251

# -- Enable Kubernetes liveness probe for Redis server

1252

enabled: false

1253

# -- Number of seconds after the container has started before [probe] is initiated

1254

initialDelaySeconds: 30

1255

# -- How often (in seconds) to perform the [probe]

1256

periodSeconds: 15

1257

# -- Number of seconds after which the [probe] times out

1258

timeoutSeconds: 15

1259

# -- Minimum consecutive successes for the [probe] to be considered successful after having failed

1260

successThreshold: 1

1261

# -- Minimum consecutive failures for the [probe] to be considered failed after having succeeded

1262

failureThreshold: 5

1263

livenessProbe:

1264

# -- Enable Kubernetes liveness probe for Redis server

1265

enabled: false

1266

# -- Number of seconds after the container has started before [probe] is initiated

1267

initialDelaySeconds: 30

1268

# -- How often (in seconds) to perform the [probe]

1269

periodSeconds: 15

1270

# -- Number of seconds after which the [probe] times out

1271

timeoutSeconds: 15

1272

# -- Minimum consecutive successes for the [probe] to be considered successful after having failed

1273

successThreshold: 1

1274

# -- Minimum consecutive failures for the [probe] to be considered failed after having succeeded

1275

failureThreshold: 5

1276

​

1277

# -- Additional containers to be added to the redis pod

1278

## Note: Supports use of custom Helm templates

1279

extraContainers: []

1280

​

1281

# -- Init containers to add to the redis pod

1282

## Note: Supports use of custom Helm templates

1283

initContainers: []

1284

​

1285

# -- Additional volumeMounts to the redis container

1286

volumeMounts: []

1287

​

1288

# -- Additional volumes to the redis pod

1289

volumes: []

1290

​

1291

# -- Annotations to be added to the Redis server Deployment

1292

deploymentAnnotations: {}

1293

​

1294

# -- Annotations to be added to the Redis server pods

1295

podAnnotations: {}

1296

​

1297

# -- Labels to be added to the Redis server pods

1298

podLabels: {}

1299

​

1300

# -- Resource limits and requests for redis

1301

resources: {}

1302

# limits:

1303

# cpu: 200m

1304

# memory: 128Mi

1305

# requests:

1306

# cpu: 100m

1307

# memory: 64Mi

1308

​

1309

# -- Redis pod-level security context

1310

# @default -- See [values.yaml]

1311

securityContext:

1312

runAsNonRoot: true

1313

runAsUser: 999

1314

seccompProfile:

1315

type: RuntimeDefault

1316

​

1317

# Redis container ports

1318

containerPorts:

1319

# -- Redis container port

1320

redis: 6379

1321

# -- Metrics container port

1322

metrics: 9121

1323

​

1324

# -- [DNS configuration]

1325

dnsConfig: {}

1326

# -- Alternative DNS policy for Redis server pods

1327

dnsPolicy: "ClusterFirst"

1328

​

1329

# -- Redis container-level security context

1330

# @default -- See [values.yaml]

1331

containerSecurityContext:

1332

readOnlyRootFilesystem: true

1333

allowPrivilegeEscalation: false

1334

capabilities:

1335

drop:

1336

- ALL

1337

​

1338

# -- Redis service port

1339

servicePort: 6379

1340

​

1341

# -- Priority class for redis pods

1342

# @default -- `""` (defaults to global.priorityClassName)

1343

priorityClassName: ""

1344

​

1345

# -- [Node selector]

1346

# @default -- `{}` (defaults to global.nodeSelector)

1347

nodeSelector: {}

1348

​

1349

# -- [Tolerations] for use with node taints

1350

# @default -- `[]` (defaults to global.tolerations)

1351

tolerations: []

1352

​

1353

# -- Assign custom [affinity] rules to the deployment

1354

# @default -- `{}` (defaults to global.affinity preset)

1355

affinity: {}

1356

​

1357

# -- Assign custom [TopologySpreadConstraints] rules to redis

1358

# @default -- `[]` (defaults to global.topologySpreadConstraints)

1359

## Ref: https://kubernetes.io/docs/concepts/workloads/pods/pod-topology-spread-constraints/

1360

## If labelSelector is left out, it will default to the labelSelector configuration of the deployment

1361

topologySpreadConstraints: []

1362

# - maxSkew: 1

1363

# topologyKey: topology.kubernetes.io/zone

1364

# whenUnsatisfiable: DoNotSchedule

1365

​

1366

# -- terminationGracePeriodSeconds for container lifecycle hook

1367

terminationGracePeriodSeconds: 30

1368

​

1369

serviceAccount:

1370

# -- Create a service account for the redis pod

1371

create: false

1372

# -- Service account name for redis pod

1373

name: ""

1374

# -- Annotations applied to created service account

1375

annotations: {}

1376

# -- Automount API credentials for the Service Account

1377

automountServiceAccountToken: false

1378

​

1379

service:

1380

# -- Redis service annotations

1381

annotations: {}

1382

# -- Additional redis service labels

1383

labels: {}

1384

​

1385

metrics:

1386

# -- Deploy metrics service

1387

enabled: false

1388

​

1389

# Redis metrics service configuration

1390

service:

1391

# -- Metrics service type

1392

type: ClusterIP

1393

# -- Metrics service clusterIP. `None` makes a "headless service" (no virtual IP)

1394

clusterIP: None

1395

# -- Metrics service annotations

1396

annotations: {}

1397

# -- Metrics service labels

1398

labels: {}

1399

# -- Metrics service port

1400

servicePort: 9121

1401

# -- Metrics service port name

1402

portName: http-metrics

1403

​

1404

serviceMonitor:

1405

# -- Enable a prometheus ServiceMonitor

1406

enabled: false

1407

# -- Interval at which metrics should be scraped

1408

interval: 30s

1409

# -- Prometheus [RelabelConfigs] to apply to samples before scraping

1410

relabelings: []

1411

# -- Prometheus [MetricRelabelConfigs] to apply to samples before ingestion

1412

metricRelabelings: []

1413

# -- Prometheus ServiceMonitor selector

1414

selector: {}

1415

# prometheus: kube-prometheus

1416

​

1417

# -- Prometheus ServiceMonitor scheme

1418

scheme: ""

1419

# -- Prometheus ServiceMonitor tlsConfig

1420

tlsConfig: {}

1421

# -- Prometheus ServiceMonitor namespace

1422

namespace: "" # "monitoring"

1423

# -- Prometheus ServiceMonitor labels

1424

additionalLabels: {}

1425

# -- Prometheus ServiceMonitor annotations

1426

annotations: {}

1427

​

1428

## Redis-HA subchart replaces custom redis deployment when `redis-ha.enabled=true`

1429

# Ref: https://github.com/DandyDeveloper/charts/blob/master/charts/redis-ha/values.yaml

1430

redis-ha:

1431

# -- Enables the Redis HA subchart and disables the custom Redis single node deployment

1432

enabled: false

1433

## Redis image

1434

image:

1435

# -- Redis repository

1436

repository: public.ecr.aws/docker/library/redis

1437

# -- Redis tag

1438

tag: 7.0.15-alpine

1439

## Prometheus redis-exporter sidecar

1440

exporter:

1441

# -- Enable Prometheus redis-exporter sidecar

1442

enabled: false

1443

# -- Repository to use for the redis-exporter

1444

image: public.ecr.aws/bitnami/redis-exporter

1445

# -- Tag to use for the redis-exporter

1446

tag: 1.57.0

1447

persistentVolume:

1448

# -- Configures persistence on Redis nodes

1449

enabled: false

1450

## Redis specific configuration options

1451

redis:

1452

# -- Redis convention for naming the cluster group: must match `^[\\w-\\.]+$` and can be templated

1453

masterGroupName: argocd

1454

# -- Any valid redis config options in this section will be applied to each server (see `redis-ha` chart)

1455

# @default -- See [values.yaml]

1456

config:

1457

# -- Will save the DB if both the given number of seconds and the given number of write operations against the DB occurred. `""` is disabled

1458

# @default -- `'""'`

1459

save: '""'

1460

## Enables a HA Proxy for better LoadBalancing / Sentinel Master support. Automatically proxies to Redis master.

1461

haproxy:

1462

# -- Enabled HAProxy LoadBalancing/Proxy

1463

enabled: true

1464

metrics:

1465

# -- HAProxy enable prometheus metric scraping

1466

enabled: true

1467

# -- Whether the haproxy pods should be forced to run on separate nodes.

1468

hardAntiAffinity: true

1469

# -- Additional affinities to add to the haproxy pods.

1470

additionalAffinities: {}

1471

# -- Assign custom [affinity] rules to the haproxy pods.

1472

affinity: |

1473

​

1474

# -- [Tolerations] for use with node taints for haproxy pods.

1475

tolerations: []

1476

# -- HAProxy container-level security context

1477

# @default -- See [values.yaml]

1478

containerSecurityContext:

1479

readOnlyRootFilesystem: true

1480

​

1481

# -- Whether the Redis server pods should be forced to run on separate nodes.

1482

hardAntiAffinity: true

1483

​

1484

# -- Additional affinities to add to the Redis server pods.

1485

additionalAffinities: {}

1486

​

1487

# -- Assign custom [affinity] rules to the Redis pods.

1488

affinity: |

1489

​

1490

# -- [Tolerations] for use with node taints for Redis pods.

1491

tolerations: []

1492

​

1493

# -- Assign custom [TopologySpreadConstraints] rules to the Redis pods.

1494

## https://kubernetes.io/docs/concepts/workloads/pods/pod-topology-spread-constraints/

1495

topologySpreadConstraints:

1496

# -- Enable Redis HA topology spread constraints

1497

enabled: false

1498

# -- Max skew of pods tolerated

1499

# @default -- `""` (defaults to `1`)

1500

maxSkew: ""

1501

# -- Topology key for spread

1502

# @default -- `""` (defaults to `topology.kubernetes.io/zone`)

1503

topologyKey: ""

1504

# -- Enforcement policy, hard or soft

1505

# @default -- `""` (defaults to `ScheduleAnyway`)

1506

whenUnsatisfiable: ""

1507

# -- Redis HA statefulset container-level security context

1508

# @default -- See [values.yaml]

1509

containerSecurityContext:

1510

readOnlyRootFilesystem: true

1511

​

1512

# External Redis parameters

1513

externalRedis:

1514

# -- External Redis server host

1515

host: ""

1516

# -- External Redis username

1517

username: ""

1518

# -- External Redis password

1519

password: ""

1520

# -- External Redis server port

1521

port: 6379

1522

# -- The name of an existing secret with Redis credentials (must contain key `redis-password`).

1523

# When it's set, the `externalRedis.password` parameter is ignored

1524

existingSecret: ""

1525

# -- External Redis Secret annotations

1526

secretAnnotations: {}

1527

​

1528

## Server

1529

server:

1530

# -- Argo CD server name

1531

name: server

1532

​

1533

# -- The number of server pods to run

1534

replicas: 1

1535

​

1536

## Argo CD server Horizontal Pod Autoscaler

1537

autoscaling:

1538

# -- Enable Horizontal Pod Autoscaler ([HPA]) for the Argo CD server

1539

enabled: false

1540

# -- Minimum number of replicas for the Argo CD server [HPA]

1541

minReplicas: 1

1542

# -- Maximum number of replicas for the Argo CD server [HPA]

1543

maxReplicas: 5

1544

# -- Average CPU utilization percentage for the Argo CD server [HPA]

1545

targetCPUUtilizationPercentage: 50

1546

# -- Average memory utilization percentage for the Argo CD server [HPA]

1547

targetMemoryUtilizationPercentage: 50

1548

# -- Configures the scaling behavior of the target in both Up and Down directions.

1549

behavior: {}

1550

# scaleDown:

1551

# stabilizationWindowSeconds: 300

1552

# policies:

1553

# - type: Pods

1554

# value: 1

1555

# periodSeconds: 180

1556

# scaleUp:

1557

# stabilizationWindowSeconds: 300

1558

# policies:

1559

# - type: Pods

1560

# value: 2

1561

# periodSeconds: 60

1562

# -- Configures custom HPA metrics for the Argo CD server

1563

# Ref: https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/

1564

metrics: []

1565

​

1566

## Argo CD server Pod Disruption Budget

1567

## Ref: https://kubernetes.io/docs/tasks/run-application/configure-pdb/

1568

pdb:

1569

# -- Deploy a [PodDisruptionBudget] for the Argo CD server

1570

enabled: false

1571

# -- Labels to be added to Argo CD server pdb

1572

labels: {}

1573

# -- Annotations to be added to Argo CD server pdb

1574

annotations: {}

1575

# -- Number of pods that are available after eviction as number or percentage (eg.: 50%)

1576

# @default -- `""` (defaults to 0 if not specified)

1577

minAvailable: ""

1578

# -- Number of pods that are unavailable after eviction as number or percentage (eg.: 50%).

1579

## Has higher precedence over `server.pdb.minAvailable`

1580

maxUnavailable: ""

1581

​

1582

## Argo CD server image

1583

image:

1584

# -- Repository to use for the Argo CD server

1585

# @default -- `""` (defaults to global.image.repository)

1586

repository: "" # defaults to global.image.repository

1587

# -- Tag to use for the Argo CD server

1588

# @default -- `""` (defaults to global.image.tag)

1589

tag: "" # defaults to global.image.tag

1590

# -- Image pull policy for the Argo CD server

1591

# @default -- `""` (defaults to global.image.imagePullPolicy)

1592

imagePullPolicy: "" # IfNotPresent

1593

​

1594

# -- Secrets with credentials to pull images from a private registry

1595

# @default -- `[]` (defaults to global.imagePullSecrets)

1596

imagePullSecrets: []

1597

​

1598

# -- Additional command line arguments to pass to Argo CD server

1599

extraArgs: []

1600

​

1601

# -- Environment variables to pass to Argo CD server

1602

env: []

1603

​

1604

# -- envFrom to pass to Argo CD server

1605

# @default -- `[]` (See [values.yaml])

1606

envFrom: []

1607

# - configMapRef:

1608

# name: config-map-name

1609

# - secretRef:

1610

# name: secret-name

1611

​

1612

# -- Specify postStart and preStop lifecycle hooks for your argo-cd-server container

1613

lifecycle: {}

1614

​

1615

## Argo CD extensions

1616

## This function in tech preview stage, do expect instability or breaking changes in newer versions.

1617

## Ref: https://github.com/argoproj-labs/argocd-extension-installer

1618

## When you enable extensions, you need to configure RBAC of logged in Argo CD user.

1619

## Ref: https://argo-cd.readthedocs.io/en/stable/operator-manual/rbac/#the-extensions-resource

1620

extensions:

1621

# -- Enable support for Argo CD extensions

1622

enabled: false

1623

​

1624

## Argo CD extension installer image

1625

image:

1626

# -- Repository to use for extension installer image

1627

repository: "quay.io/argoprojlabs/argocd-extension-installer"

1628

# -- Tag to use for extension installer image

1629

tag: "v0.0.1"

1630

# -- Image pull policy for extensions

1631

# @default -- `""` (defaults to global.image.imagePullPolicy)

1632

imagePullPolicy: ""

1633

​

1634

# -- Extensions for Argo CD

1635

# @default -- `[]` (See [values.yaml])

1636

## Ref: https://github.com/argoproj-labs/argocd-extension-metrics#install-ui-extension

1637

extensionList: []

1638

# - name: extension-metrics

1639

# env:

1640

# - name: EXTENSION\_URL

1641

# value: https://github.com/argoproj-labs/argocd-extension-metrics/releases/download/v1.0.0/extension.tar.gz

1642

# - name: EXTENSION\_CHECKSUM\_URL

1643

# value: https://github.com/argoproj-labs/argocd-extension-metrics/releases/download/v1.0.0/extension\_checksums.txt

1644

​

1645

# -- Server UI extensions container-level security context

1646

# @default -- See [values.yaml]

1647

containerSecurityContext:

1648

runAsNonRoot: true

1649

readOnlyRootFilesystem: true

1650

allowPrivilegeEscalation: false

1651

runAsUser: 1000

1652

seccompProfile:

1653

type: RuntimeDefault

1654

capabilities:

1655

drop:

1656

- ALL

1657

​

1658

# -- Resource limits and requests for the argocd-extensions container

1659

resources: {}

1660

# limits:

1661

# cpu: 50m

1662

# memory: 128Mi

1663

# requests:

1664

# cpu: 10m

1665

# memory: 64Mi

1666

​

1667

# -- Additional containers to be added to the server pod

1668

## Note: Supports use of custom Helm templates

1669

extraContainers: []

1670

# - name: my-sidecar

1671

# image: nginx:latest

1672

# - name: lemonldap-ng-controller

1673

# image: lemonldapng/lemonldap-ng-controller:0.2.0

1674

# args:

1675

# - /lemonldap-ng-controller

1676

# - --alsologtostderr

1677

# - --configmap=$(POD\_NAMESPACE)/lemonldap-ng-configuration

1678

# env:

1679

# - name: POD\_NAME

1680

# valueFrom:

1681

# fieldRef:

1682

# fieldPath: metadata.name

1683

# - name: POD\_NAMESPACE

1684

# valueFrom:

1685

# fieldRef:

1686

# fieldPath: metadata.namespace

1687

# volumeMounts:

1688

# - name: copy-portal-skins

1689

# mountPath: /srv/var/lib/lemonldap-ng/portal/skins

1690

​

1691

# -- Init containers to add to the server pod

1692

## If your target Kubernetes cluster(s) require a custom credential (exec) plugin

1693

## you could use this (and the same in the application controller pod) to provide such executable

1694

## Ref: https://kubernetes.io/docs/reference/access-authn-authz/authentication/#client-go-credential-plugins

1695

initContainers: []

1696

# - name: download-tools

1697

# image: alpine:3

1698

# command: [sh, -c]

1699

# args:

1700

# - wget -qO kubelogin.zip https://github.com/Azure/kubelogin/releases/download/v0.0.25/kubelogin-linux-amd64.zip &&

1701

# unzip kubelogin.zip && mv bin/linux\_amd64/kubelogin /custom-tools/

1702

# volumeMounts:

1703

# - mountPath: /custom-tools

1704

# name: custom-tools

1705

​

1706

# -- Additional volumeMounts to the server main container

1707

volumeMounts: []

1708

# - mountPath: /usr/local/bin/kubelogin

1709

# name: custom-tools

1710

# subPath: kubelogin

1711

​

1712

# -- Additional volumes to the server pod

1713

volumes: []

1714

# - name: custom-tools

1715

# emptyDir: {}

1716

​

1717

# -- Annotations to be added to server Deployment

1718

deploymentAnnotations: {}

1719

​

1720

# -- Annotations to be added to server pods

1721

podAnnotations: {}

1722

​

1723

# -- Labels to be added to server pods

1724

podLabels: {}

1725

​

1726

# -- Resource limits and requests for the Argo CD server

1727

resources: {}

1728

# limits:

1729

# cpu: 100m

1730

# memory: 128Mi

1731

# requests:

1732

# cpu: 50m

1733

# memory: 64Mi

1734

​

1735

# Server container ports

1736

containerPorts:

1737

# -- Server container port

1738

server: 8080

1739

# -- Metrics container port

1740

metrics: 8083

1741

​

1742

# -- Host Network for Server pods

1743

hostNetwork: false

1744

​

1745

# -- [DNS configuration]

1746

dnsConfig: {}

1747

# -- Alternative DNS policy for Server pods

1748

dnsPolicy: "ClusterFirst"

1749

​

1750

# -- Server container-level security context

1751

# @default -- See [values.yaml]

1752

containerSecurityContext:

1753

runAsNonRoot: true

1754

readOnlyRootFilesystem: true

1755

allowPrivilegeEscalation: false

1756

seccompProfile:

1757

type: RuntimeDefault

1758

capabilities:

1759

drop:

1760

- ALL

1761

​

1762

## Readiness and liveness probes for default backend

1763

## Ref: https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-probes/

1764

readinessProbe:

1765

# -- Minimum consecutive failures for the [probe] to be considered failed after having succeeded

1766

failureThreshold: 3

1767

# -- Number of seconds after the container has started before [probe] is initiated

1768

initialDelaySeconds: 10

1769

# -- How often (in seconds) to perform the [probe]

1770

periodSeconds: 10

1771

# -- Minimum consecutive successes for the [probe] to be considered successful after having failed

1772

successThreshold: 1

1773

# -- Number of seconds after which the [probe] times out

1774

timeoutSeconds: 1

1775

​

1776

livenessProbe:

1777

# -- Minimum consecutive failures for the [probe] to be considered failed after having succeeded

1778

failureThreshold: 3

1779

# -- Number of seconds after the container has started before [probe] is initiated

1780

initialDelaySeconds: 10

1781

# -- How often (in seconds) to perform the [probe]

1782

periodSeconds: 10

1783

# -- Minimum consecutive successes for the [probe] to be considered successful after having failed

1784

successThreshold: 1

1785

# -- Number of seconds after which the [probe] times out

1786

timeoutSeconds: 1

1787

​

1788

# -- terminationGracePeriodSeconds for container lifecycle hook

1789

terminationGracePeriodSeconds: 30

1790

​

1791

# -- Priority class for the Argo CD server pods

1792

# @default -- `""` (defaults to global.priorityClassName)

1793

priorityClassName: ""

1794

​

1795

# -- [Node selector]

1796

# @default -- `{}` (defaults to global.nodeSelector)

1797

nodeSelector: {}

1798

​

1799

# -- [Tolerations] for use with node taints

1800

# @default -- `[]` (defaults to global.tolerations)

1801

tolerations: []

1802

​

1803

# -- Assign custom [affinity] rules to the deployment

1804

# @default -- `{}` (defaults to global.affinity preset)

1805

affinity: {}

1806

​

1807

# -- Assign custom [TopologySpreadConstraints] rules to the Argo CD server

1808

# @default -- `[]` (defaults to global.topologySpreadConstraints)

1809

## Ref: https://kubernetes.io/docs/concepts/workloads/pods/pod-topology-spread-constraints/

1810

## If labelSelector is left out, it will default to the labelSelector configuration of the deployment

1811

topologySpreadConstraints: []

1812

# - maxSkew: 1

1813

# topologyKey: topology.kubernetes.io/zone

1814

# whenUnsatisfiable: DoNotSchedule

1815

​

1816

# -- Deployment strategy to be added to the server Deployment

1817

deploymentStrategy: {}

1818

# type: RollingUpdate

1819

# rollingUpdate:

1820

# maxSurge: 25%

1821

# maxUnavailable: 25%

1822

​

1823

# TLS certificate configuration via cert-manager

1824

## Ref: https://argo-cd.readthedocs.io/en/stable/operator-manual/tls/#tls-certificates-used-by-argocd-server

1825

certificate:

1826

# -- Deploy a Certificate resource (requires cert-manager)

1827

enabled: false

1828

# -- The name of the Secret that will be automatically created and managed by this Certificate resource

1829

secretName: argocd-server-tls

1830

# -- Certificate primary domain (commonName)

1831

domain: argocd.example.com

1832

# -- Certificate Subject Alternate Names (SANs)

1833

additionalHosts: []

1834

# -- The requested 'duration' (i.e. lifetime) of the certificate.

1835

# @default -- `""` (defaults to 2160h = 90d if not specified)

1836

## Ref: https://cert-manager.io/docs/usage/certificate/#renewal

1837

duration: ""

1838

# -- How long before the expiry a certificate should be renewed.

1839

# @default -- `""` (defaults to 360h = 15d if not specified)

1840

## Ref: https://cert-manager.io/docs/usage/certificate/#renewal

1841

renewBefore: ""

1842

# Certificate issuer

1843

## Ref: https://cert-manager.io/docs/concepts/issuer

1844

issuer:

1845

# -- Certificate issuer group. Set if using an external issuer. Eg. `cert-manager.io`

1846

group: ""

1847

# -- Certificate issuer kind. Either `Issuer` or `ClusterIssuer`

1848

kind: ""

1849

# -- Certificate issuer name. Eg. `letsencrypt`

1850

name: ""

1851

# Private key of the certificate

1852

privateKey:

1853

# -- Rotation policy of private key when certificate is re-issued. Either: `Never` or `Always`

1854

rotationPolicy: Never

1855

# -- The private key cryptography standards (PKCS) encoding for private key. Either: `PCKS1` or `PKCS8`

1856

encoding: PKCS1

1857

# -- Algorithm used to generate certificate private key. One of: `RSA`, `Ed25519` or `ECDSA`

1858

algorithm: RSA

1859

# -- Key bit size of the private key. If algorithm is set to `Ed25519`, size is ignored.

1860

size: 2048

1861

# -- Annotations to be applied to the Server Certificate

1862

annotations: {}

1863

# -- Usages for the certificate

1864

### Ref: https://cert-manager.io/docs/reference/api-docs/#cert-manager.io/v1.KeyUsage

1865

usages: []

1866

​

1867

# TLS certificate configuration via Secret

1868

## Ref: https://argo-cd.readthedocs.io/en/stable/operator-manual/tls/#tls-certificates-used-by-argocd-server

1869

certificateSecret:

1870

# -- Create argocd-server-tls secret

1871

enabled: false

1872

# -- Annotations to be added to argocd-server-tls secret

1873

annotations: {}

1874

# -- Labels to be added to argocd-server-tls secret

1875

labels: {}

1876

# -- Private Key of the certificate

1877

key: ''

1878

# -- Certificate data

1879

crt: ''

1880

​

1881

## Server service configuration

1882

service:

1883

# -- Server service annotations

1884

annotations: {}

1885

# -- Server service labels

1886

labels: {}

1887

# -- Server service type

1888

type: ClusterIP

1889

# -- Server service http port for NodePort service type (only if `server.service.type` is set to "NodePort")

1890

nodePortHttp: 30080

1891

# -- Server service https port for NodePort service type (only if `server.service.type` is set to "NodePort")

1892

nodePortHttps: 30443

1893

# -- Server service http port

1894

servicePortHttp: 80

1895

# -- Server service https port

1896

servicePortHttps: 443

1897

# -- Server service http port name, can be used to route traffic via istio

1898

servicePortHttpName: http

1899

# -- Server service https port name, can be used to route traffic via istio

1900

servicePortHttpsName: https

1901

# -- Server service https port appProtocol. (should be upper case - i.e. HTTPS)

1902

# servicePortHttpsAppProtocol: HTTPS

1903

# -- LoadBalancer will get created with the IP specified in this field

1904

loadBalancerIP: ""

1905

# -- Source IP ranges to allow access to service from

1906

loadBalancerSourceRanges: []

1907

# -- Server service external IPs

1908

externalIPs: []

1909

# -- Denotes if this Service desires to route external traffic to node-local or cluster-wide endpoints

1910

externalTrafficPolicy: ""

1911

# -- Used to maintain session affinity. Supports `ClientIP` and `None`

1912

sessionAffinity: ""

1913

​

1914

## Server metrics service configuration

1915

metrics:

1916

# -- Deploy metrics service

1917

enabled: false

1918

service:

1919

# -- Metrics service type

1920

type: ClusterIP

1921

# -- Metrics service clusterIP. `None` makes a "headless service" (no virtual IP)

1922

clusterIP: ""

1923

# -- Metrics service annotations

1924

annotations: {}

1925

# -- Metrics service labels

1926

labels: {}

1927

# -- Metrics service port

1928

servicePort: 8083

1929

# -- Metrics service port name

1930

portName: http-metrics

1931

serviceMonitor:

1932

# -- Enable a prometheus ServiceMonitor

1933

enabled: false

1934

# -- Prometheus ServiceMonitor interval

1935

interval: 30s

1936

# -- Prometheus ServiceMonitor scrapeTimeout. If empty, Prometheus uses the global scrape timeout unless it is less than the target's scrape interval value in which the latter is used.

1937

scrapeTimeout: ""

1938

# -- Prometheus [RelabelConfigs] to apply to samples before scraping

1939

relabelings: []

1940

# -- Prometheus [MetricRelabelConfigs] to apply to samples before ingestion

1941

metricRelabelings: []

1942

# -- Prometheus ServiceMonitor selector

1943

selector: {}

1944

# prometheus: kube-prometheus

1945

​

1946

# -- Prometheus ServiceMonitor scheme

1947

scheme: ""

1948

# -- Prometheus ServiceMonitor tlsConfig

1949

tlsConfig: {}

1950

# -- Prometheus ServiceMonitor namespace

1951

namespace: "" # monitoring

1952

# -- Prometheus ServiceMonitor labels

1953

additionalLabels: {}

1954

# -- Prometheus ServiceMonitor annotations

1955

annotations: {}

1956

​

1957

serviceAccount:

1958

# -- Create server service account

1959

create: true

1960

# -- Server service account name

1961

name: argocd-server

1962

# -- Annotations applied to created service account

1963

annotations: {}

1964

# -- Labels applied to created service account

1965

labels: {}

1966

# -- Automount API credentials for the Service Account

1967

automountServiceAccountToken: true

1968

​

1969

# Argo CD server ingress configuration

1970

ingress:

1971

# -- Enable an ingress resource for the Argo CD server

1972

enabled: false

1973

# -- Specific implementation for ingress controller. One of `generic`, `aws` or `gke`

1974

## Additional configuration might be required in related configuration sections

1975

controller: generic

1976

# -- Additional ingress labels

1977

labels: {}

1978

# -- Additional ingress annotations

1979

## Ref: https://argo-cd.readthedocs.io/en/stable/operator-manual/ingress/#option-1-ssl-passthrough

1980

annotations: {}

1981

# nginx.ingress.kubernetes.io/force-ssl-redirect: "true"

1982

# nginx.ingress.kubernetes.io/ssl-passthrough: "true"

1983

​

1984

# -- Defines which ingress controller will implement the resource

1985

ingressClassName: ""

1986

​

1987

# -- Argo CD server hostname

1988

## NOTE: Hostname must be provided if Ingress is enabled

1989

hostname: argocd.example.com

1990

​

1991

# -- The path to Argo CD server

1992

path: /

1993

​

1994

# -- Ingress path type. One of `Exact`, `Prefix` or `ImplementationSpecific`

1995

pathType: Prefix

1996

​

1997

# -- Enable TLS configuration for the hostname defined at `server.ingress.hostname`

1998

## TLS certificate will be retrieved from a TLS secret `argocd-server-tls`

1999

## You can create this secret via `certificate` or `certificateSecret` option

2000

tls: false

2001

​

2002

# -- The list of additional hostnames to be covered by ingress record

2003

# @default -- `[]` (See [values.yaml])

2004

extraHosts: []

2005

# - name: argocd.example.com

2006

# path: /

2007

​

2008

# -- Additional ingress paths

2009

# @default -- `[]` (See [values.yaml])

2010

extraPaths: []

2011

# - path: /\*

2012

# pathType: Prefix

2013

# backend:

2014

# service:

2015

# name: ssl-redirect

2016

# port:

2017

# name: use-annotation

2018

​

2019

# -- Additional ingress rules

2020

# @default -- `[]` (See [values.yaml])

2021

extraRules: []

2022

# - host: example.example.com

2023

# http:

2024

# path: /

2025

# backend:

2026

# service:

2027

# name: example-svc

2028

# port:

2029

# name: http

2030

​

2031

# -- Additional TLS configuration

2032

# @default -- `[]` (See [values.yaml])

2033

extraTls: []

2034

# - hosts:

2035

# - argocd.example.com

2036

# secretName: your-certificate-name

2037

​

2038

# AWS specific options for Application Load Balancer

2039

# Applies only when `serv.ingress.controller` is set to `aws`

2040

## Ref: https://argo-cd.readthedocs.io/en/stable/operator-manual/ingress/#aws-application-load-balancers-albs-and-classic-elb-http-mode

2041

aws:

2042

# -- Backend protocol version for the AWS ALB gRPC service

2043

## This tells AWS to send traffic from the ALB using HTTP2. Can use gRPC as well if you want to leverage gRPC specific features

2044

backendProtocolVersion: HTTP2

2045

# -- Service type for the AWS ALB gRPC service

2046

## Can be of type NodePort or ClusterIP depending on which mode you are running.

2047

## Instance mode needs type NodePort, IP mode needs type ClusterIP

2048

## Ref: https://kubernetes-sigs.github.io/aws-load-balancer-controller/v2.2/how-it-works/#ingress-traffic

2049

serviceType: NodePort

2050

​

2051

# Google specific options for Google Application Load Balancer

2052

# Applies only when `server.ingress.controller` is set to `gke`

2053

## Ref: https://argo-cd.readthedocs.io/en/stable/operator-manual/ingress/#google-cloud-load-balancers-with-kubernetes-ingress

2054

gke:

2055

# -- Google [BackendConfig] resource, for use with the GKE Ingress Controller

2056

# @default -- `{}` (See [values.yaml])

2057

## Ref: https://cloud.google.com/kubernetes-engine/docs/how-to/ingress-features#configuring\_ingress\_features\_through\_frontendconfig\_parameters

2058

backendConfig: {}

2059

# iap:

2060

# enabled: true

2061

# oauthclientCredentials:

2062

# secretName: argocd-secret

2063

​

2064

# -- Google [FrontendConfig] resource, for use with the GKE Ingress Controller

2065

# @default -- `{}` (See [values.yaml])

2066

## Ref: https://cloud.google.com/kubernetes-engine/docs/how-to/ingress-features#configuring\_ingress\_features\_through\_frontendconfig\_parameters

2067

frontendConfig: {}

2068

# redirectToHttps:

2069

# enabled: true

2070

# responseCodeName: RESPONSE\_CODE

2071

​

2072

# Managed GKE certificate for ingress hostname

2073

managedCertificate:

2074

# -- Create ManagedCertificate resource and annotations for Google Load balancer

2075

## Ref: https://cloud.google.com/kubernetes-engine/docs/how-to/managed-certs

2076

create: true

2077

# -- Additional domains for ManagedCertificate resource

2078

extraDomains: []

2079

# - argocd.example.com

2080

​

2081

# Dedicated gRPC ingress for ingress controllers that supports only single backend protocol per Ingress resource

2082

# Ref: https://argo-cd.readthedocs.io/en/stable/operator-manual/ingress/#option-2-multiple-ingress-objects-and-hosts

2083

ingressGrpc:

2084

# -- Enable an ingress resource for the Argo CD server for dedicated [gRPC-ingress]

2085

enabled: false

2086

# -- Additional ingress annotations for dedicated [gRPC-ingress]

2087

annotations: {}

2088

# -- Additional ingress labels for dedicated [gRPC-ingress]

2089

labels: {}

2090

# -- Defines which ingress controller will implement the resource [gRPC-ingress]

2091

ingressClassName: ""

2092

​

2093

# -- Argo CD server hostname for dedicated [gRPC-ingress]

2094

# @default -- `""` (defaults to grpc.`server.ingress.hostname`)

2095

hostname: ""

2096

​

2097

# -- Argo CD server ingress path for dedicated [gRPC-ingress]

2098

path: /

2099

​

2100

# -- Ingress path type for dedicated [gRPC-ingress]. One of `Exact`, `Prefix` or `ImplementationSpecific`

2101

pathType: Prefix

2102

​

2103

# -- Enable TLS configuration for the hostname defined at `server.ingressGrpc.hostname`

2104

## TLS certificate will be retrieved from a TLS secret with name: `argocd-server-grpc-tls`

2105

tls: false

2106

​

2107

# -- The list of additional hostnames to be covered by ingress record

2108

# @default -- `[]` (See [values.yaml])

2109

extraHosts: []

2110

# - name: grpc.argocd.example.com

2111

# path: /

2112

​

2113

# -- Additional ingress paths for dedicated [gRPC-ingress]

2114

# @default -- `[]` (See [values.yaml])

2115

extraPaths: []

2116

# - path: /\*

2117

# pathType: Prefix

2118

# backend:

2119

# service:

2120

# name: ssl-redirect

2121

# port:

2122

# name: use-annotation

2123

​

2124

# -- Additional ingress rules

2125

# @default -- `[]` (See [values.yaml])

2126

extraRules: []

2127

# - host: example.example.com

2128

# http:

2129

# path: /

2130

# backend:

2131

# service:

2132

# name: example-svc

2133

# port:

2134

# name: http

2135

​

2136

# -- Additional TLS configuration for dedicated [gRPC-ingress]

2137

# @default -- `[]` (See [values.yaml])

2138

extraTls: []

2139

# - secretName: your-certificate-name

2140

# hosts:

2141

# - argocd.example.com

2142

​

2143

# Create a OpenShift Route with SSL passthrough for UI and CLI

2144

# Consider setting 'hostname' e.g. https://argocd.apps-crc.testing/ using your Default Ingress Controller Domain

2145

# Find your domain with: kubectl describe --namespace=openshift-ingress-operator ingresscontroller/default | grep Domain:

2146

# If 'hostname' is an empty string "" OpenShift will create a hostname for you.

2147

route:

2148

# -- Enable an OpenShift Route for the Argo CD server

2149

enabled: false

2150

# -- Openshift Route annotations

2151

annotations: {}

2152

# -- Hostname of OpenShift Route

2153

hostname: ""

2154

# -- Termination type of Openshift Route

2155

termination\_type: passthrough

2156

# -- Termination policy of Openshift Route

2157

termination\_policy: None

2158

​

2159

## Repo Server

2160

repoServer:

2161

# -- Repo server name

2162

name: repo-server

2163

​

2164

# -- The number of repo server pods to run

2165

replicas: 1

2166

​

2167

## Repo server Horizontal Pod Autoscaler

2168

autoscaling:

2169

# -- Enable Horizontal Pod Autoscaler ([HPA]) for the repo server

2170

enabled: false

2171

# -- Minimum number of replicas for the repo server [HPA]

2172

minReplicas: 1

2173

# -- Maximum number of replicas for the repo server [HPA]

2174

maxReplicas: 5

2175

# -- Average CPU utilization percentage for the repo server [HPA]

2176

targetCPUUtilizationPercentage: 50

2177

# -- Average memory utilization percentage for the repo server [HPA]

2178

targetMemoryUtilizationPercentage: 50

2179

# -- Configures the scaling behavior of the target in both Up and Down directions.

2180

behavior: {}

2181

# scaleDown:

2182

# stabilizationWindowSeconds: 300

2183

# policies:

2184

# - type: Pods

2185

# value: 1

2186

# periodSeconds: 180

2187

# scaleUp:

2188

# stabilizationWindowSeconds: 300

2189

# policies:

2190

# - type: Pods

2191

# value: 2

2192

# periodSeconds: 60

2193

# -- Configures custom HPA metrics for the Argo CD repo server

2194

# Ref: https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale/

2195

metrics: []

2196

​

2197

## Repo server Pod Disruption Budget

2198

## Ref: https://kubernetes.io/docs/tasks/run-application/configure-pdb/

2199

pdb:

2200

# -- Deploy a [PodDisruptionBudget] for the repo server

2201

enabled: false

2202

# -- Labels to be added to repo server pdb

2203

labels: {}

2204

# -- Annotations to be added to repo server pdb

2205

annotations: {}

2206

# -- Number of pods that are available after eviction as number or percentage (eg.: 50%)

2207

# @default -- `""` (defaults to 0 if not specified)

2208

minAvailable: ""

2209

# -- Number of pods that are unavailable after eviction as number or percentage (eg.: 50%).

2210

## Has higher precedence over `repoServer.pdb.minAvailable`

2211

maxUnavailable: ""

2212

​

2213

## Repo server image

2214

image:

2215

# -- Repository to use for the repo server

2216

# @default -- `""` (defaults to global.image.repository)

2217

repository: ""

2218

# -- Tag to use for the repo server

2219

# @default -- `""` (defaults to global.image.tag)

2220

tag: ""

2221

# -- Image pull policy for the repo server

2222

# @default -- `""` (defaults to global.image.imagePullPolicy)

2223

imagePullPolicy: ""

2224

​

2225

# -- Secrets with credentials to pull images from a private registry

2226

# @default -- `[]` (defaults to global.imagePullSecrets)

2227

imagePullSecrets: []

2228

​

2229

# -- Additional command line arguments to pass to repo server

2230

extraArgs: []

2231

​

2232

# -- Environment variables to pass to repo server

2233

env: []

2234

​

2235

# -- envFrom to pass to repo server

2236

# @default -- `[]` (See [values.yaml])

2237

envFrom: []

2238

# - configMapRef:

2239

# name: config-map-name

2240

# - secretRef:

2241

# name: secret-name

2242

​

2243

# -- Specify postStart and preStop lifecycle hooks for your argo-repo-server container

2244

lifecycle: {}

2245

​

2246

# -- Additional containers to be added to the repo server pod

2247

## Ref: https://argo-cd.readthedocs.io/en/stable/user-guide/config-management-plugins/

2248

## Note: Supports use of custom Helm templates

2249

extraContainers: []

2250

# - name: cmp-my-plugin

2251

# command:

2252

# - "/var/run/argocd/argocd-cmp-server"

2253

# image: busybox

2254

# securityContext:

2255

# runAsNonRoot: true

2256

# runAsUser: 999

2257

# volumeMounts:

2258

# - mountPath: /var/run/argocd

2259

# name: var-files

2260

# - mountPath: /home/argocd/cmp-server/plugins

2261

# name: plugins

2262

# # Remove this volumeMount if you've chosen to bake the config file into the sidecar image.

2263

# - mountPath: /home/argocd/cmp-server/config/plugin.yaml

2264

# subPath: my-plugin.yaml

2265

# name: argocd-cmp-cm

2266

# # Starting with v2.4, do NOT mount the same tmp volume as the repo-server container. The filesystem separation helps

2267

# # mitigate path traversal attacks.

2268

# - mountPath: /tmp

2269

# name: cmp-tmp

2270

# - name: cmp-my-plugin2

2271

# command:

2272

# - "/var/run/argocd/argocd-cmp-server"

2273

# image: busybox

2274

# securityContext:

2275

# runAsNonRoot: true

2276

# runAsUser: 999

2277

# volumeMounts:

2278

# - mountPath: /var/run/argocd

2279

# name: var-files

2280

# # Remove this volumeMount if you've chosen to bake the config file into the sidecar image.

2281

# - mountPath: /home/argocd/cmp-server/plugins

2282

# name: plugins

2283

# - mountPath: /home/argocd/cmp-server/config/plugin.yaml

2284

# subPath: my-plugin2.yaml

2285

# name: argocd-cmp-cm

2286

# # Starting with v2.4, do NOT mount the same tmp volume as the repo-server container. The filesystem separation helps

2287

# # mitigate path traversal attacks.

2288

# - mountPath: /tmp

2289

# name: cmp-tmp

2290

​

2291

# -- Init containers to add to the repo server pods

2292

initContainers: []

2293

​

2294

# -- Additional volumeMounts to the repo server main container

2295

volumeMounts: []

2296

​

2297

# -- Additional volumes to the repo server pod

2298

volumes: []

2299

# - name: argocd-cmp-cm

2300

# configMap:

2301

# name: argocd-cmp-cm

2302

# - name: cmp-tmp

2303

# emptyDir: {}

2304

​

2305

# -- Volumes to be used in replacement of emptydir on default volumes

2306

existingVolumes: {}

2307

# gpgKeyring:

2308

# persistentVolumeClaim:

2309

# claimName: pvc-argocd-repo-server-keyring

2310

# helmWorkingDir:

2311

# persistentVolumeClaim:

2312

# claimName: pvc-argocd-repo-server-workdir

2313

# tmp:

2314

# persistentVolumeClaim:

2315

# claimName: pvc-argocd-repo-server-tmp

2316

# varFiles:

2317

# persistentVolumeClaim:

2318

# claimName: pvc-argocd-repo-server-varfiles

2319

# plugins:

2320

# persistentVolumeClaim:

2321

# claimName: pvc-argocd-repo-server-plugins

2322

​

2323

# -- Toggle the usage of a ephemeral Helm working directory

2324

useEphemeralHelmWorkingDir: true

2325

​

2326

# -- Annotations to be added to repo server Deployment

2327

deploymentAnnotations: {}

2328

​

2329

# -- Annotations to be added to repo server pods

2330

podAnnotations: {}

2331

​

2332

# -- Labels to be added to repo server pods

2333

podLabels: {}

2334

​

2335

# -- Resource limits and requests for the repo server pods

2336

resources: {}

2337

# limits:

2338

# cpu: 50m

2339

# memory: 128Mi

2340

# requests:

2341

# cpu: 10m

2342

# memory: 64Mi

2343

​

2344

# Repo server container ports

2345

containerPorts:

2346

# -- Repo server container port

2347

server: 8081

2348

# -- Metrics container port

2349

metrics: 8084

2350

​

2351

# -- Host Network for Repo server pods

2352

hostNetwork: false

2353

​

2354

# -- [DNS configuration]

2355

dnsConfig: {}

2356

# -- Alternative DNS policy for Repo server pods

2357

dnsPolicy: "ClusterFirst"

2358

​

2359

# -- Repo server container-level security context

2360

# @default -- See [values.yaml]

2361

containerSecurityContext:

2362

runAsNonRoot: true

2363

readOnlyRootFilesystem: true

2364

allowPrivilegeEscalation: false

2365

seccompProfile:

2366

type: RuntimeDefault

2367

capabilities:

2368

drop:

2369

- ALL

2370

​

2371

## Readiness and liveness probes for default backend

2372

## Ref: https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-probes/

2373

readinessProbe:

2374

# -- Minimum consecutive failures for the [probe] to be considered failed after having succeeded

2375

failureThreshold: 3

2376

# -- Number of seconds after the container has started before [probe] is initiated

2377

initialDelaySeconds: 10

2378

# -- How often (in seconds) to perform the [probe]

2379

periodSeconds: 10

2380

# -- Minimum consecutive successes for the [probe] to be considered successful after having failed

2381

successThreshold: 1

2382

# -- Number of seconds after which the [probe] times out

2383

timeoutSeconds: 1

2384

​

2385

livenessProbe:

2386

# -- Minimum consecutive failures for the [probe] to be considered failed after having succeeded

2387

failureThreshold: 3

2388

# -- Number of seconds after the container has started before [probe] is initiated

2389

initialDelaySeconds: 10

2390

# -- How often (in seconds) to perform the [probe]

2391

periodSeconds: 10

2392

# -- Minimum consecutive successes for the [probe] to be considered successful after having failed

2393

successThreshold: 1

2394

# -- Number of seconds after which the [probe] times out

2395

timeoutSeconds: 1

2396

​

2397

# -- terminationGracePeriodSeconds for container lifecycle hook

2398

terminationGracePeriodSeconds: 30

2399

​

2400

# -- [Node selector]

2401

# @default -- `{}` (defaults to global.nodeSelector)

2402

nodeSelector: {}

2403

​

2404

# -- [Tolerations] for use with node taints

2405

# @default -- `[]` (defaults to global.tolerations)

2406

tolerations: []

2407

​

2408

# -- Assign custom [affinity] rules to the deployment

2409

# @default -- `{}` (defaults to global.affinity preset)

2410

affinity: {}

2411

​

2412

# -- Assign custom [TopologySpreadConstraints] rules to the repo server

2413

# @default -- `[]` (defaults to global.topologySpreadConstraints)

2414

## Ref: https://kubernetes.io/docs/concepts/workloads/pods/pod-topology-spread-constraints/

2415

## If labelSelector is left out, it will default to the labelSelector configuration of the deployment

2416

topologySpreadConstraints: []

2417

# - maxSkew: 1

2418

# topologyKey: topology.kubernetes.io/zone

2419

# whenUnsatisfiable: DoNotSchedule

2420

​

2421

# -- Deployment strategy to be added to the repo server Deployment

2422

deploymentStrategy: {}

2423

# type: RollingUpdate

2424

# rollingUpdate:

2425

# maxSurge: 25%

2426

# maxUnavailable: 25%

2427

​

2428

# -- Priority class for the repo server pods

2429

# @default -- `""` (defaults to global.priorityClassName)

2430

priorityClassName: ""

2431

​

2432

# TLS certificate configuration via Secret

2433

## Ref: https://argo-cd.readthedocs.io/en/stable/operator-manual/tls/#configuring-tls-to-argocd-repo-server

2434

## Note: Issuing certificates via cert-manager in not supported right now because it's not possible to restart repo server automatically without extra controllers.

2435

certificateSecret:

2436

# -- Create argocd-repo-server-tls secret

2437

enabled: false

2438

# -- Annotations to be added to argocd-repo-server-tls secret

2439

annotations: {}

2440

# -- Labels to be added to argocd-repo-server-tls secret

2441

labels: {}

2442

# -- Certificate authority. Required for self-signed certificates.

2443

ca: ''

2444

# -- Certificate private key

2445

key: ''

2446

# -- Certificate data. Must contain SANs of Repo service (ie: argocd-repo-server, argocd-repo-server.argo-cd.svc)

2447

crt: ''

2448

​

2449

## Repo server service configuration

2450

service:

2451

# -- Repo server service annotations

2452

annotations: {}

2453

# -- Repo server service labels

2454

labels: {}

2455

# -- Repo server service port

2456

port: 8081

2457

# -- Repo server service port name

2458

portName: tcp-repo-server

2459

​

2460

## Repo server metrics service configuration

2461

metrics:

2462

# -- Deploy metrics service

2463

enabled: false

2464

service:

2465

# -- Metrics service type

2466

type: ClusterIP

2467

# -- Metrics service clusterIP. `None` makes a "headless service" (no virtual IP)

2468

clusterIP: ""

2469

# -- Metrics service annotations

2470

annotations: {}

2471

# -- Metrics service labels

2472

labels: {}

2473

# -- Metrics service port

2474

servicePort: 8084

2475

# -- Metrics service port name

2476

portName: http-metrics

2477

serviceMonitor:

2478

# -- Enable a prometheus ServiceMonitor

2479

enabled: false

2480

# -- Prometheus ServiceMonitor interval

2481

interval: 30s

2482

# -- Prometheus ServiceMonitor scrapeTimeout. If empty, Prometheus uses the global scrape timeout unless it is less than the target's scrape interval value in which the latter is used.

2483

scrapeTimeout: ""

2484

# -- Prometheus [RelabelConfigs] to apply to samples before scraping

2485

relabelings: []

2486

# -- Prometheus [MetricRelabelConfigs] to apply to samples before ingestion

2487

metricRelabelings: []

2488

# -- Prometheus ServiceMonitor selector

2489

selector: {}

2490

# prometheus: kube-prometheus

2491

​

2492

# -- Prometheus ServiceMonitor scheme

2493

scheme: ""

2494

# -- Prometheus ServiceMonitor tlsConfig

2495

tlsConfig: {}

2496

# -- Prometheus ServiceMonitor namespace

2497

namespace: "" # "monitoring"

2498

# -- Prometheus ServiceMonitor labels

2499

additionalLabels: {}

2500

# -- Prometheus ServiceMonitor annotations

2501

annotations: {}

2502

​

2503

## Enable Custom Rules for the Repo server's Cluster Role resource

2504

## Enable this and set the rules: to whatever custom rules you want for the Cluster Role resource.

2505

## Defaults to off

2506

clusterRoleRules:

2507

# -- Enable custom rules for the Repo server's Cluster Role resource

2508

enabled: false

2509

# -- List of custom rules for the Repo server's Cluster Role resource

2510

rules: []

2511

​

2512

## Repo server service account

2513

## If create is set to true, make sure to uncomment the name and update the rbac section below

2514

serviceAccount:

2515

# -- Create repo server service account

2516

create: true

2517

# -- Repo server service account name

2518

name: "" # "argocd-repo-server"

2519

# -- Annotations applied to created service account

2520

annotations: {}

2521

# -- Labels applied to created service account

2522

labels: {}

2523

# -- Automount API credentials for the Service Account

2524

automountServiceAccountToken: true

2525

​

2526

# -- Repo server rbac rules

2527

rbac: []

2528

# - apiGroups:

2529

# - argoproj.io

2530

# resources:

2531

# - applications

2532

# verbs:

2533

# - get

2534

# - list

2535

# - watch

2536

​

2537

## ApplicationSet controller

2538

applicationSet:

2539

# -- Enable ApplicationSet controller

2540

enabled: true

2541

​

2542

# -- ApplicationSet controller name string

2543

name: applicationset-controller

2544

​

2545

# -- The number of ApplicationSet controller pods to run

2546

replicas: 1

2547

​

2548

## ApplicationSet controller Pod Disruption Budget

2549

## Ref: https://kubernetes.io/docs/tasks/run-application/configure-pdb/

2550

pdb:

2551

# -- Deploy a [PodDisruptionBudget] for the ApplicationSet controller

2552

enabled: false

2553

# -- Labels to be added to ApplicationSet controller pdb

2554

labels: {}

2555

# -- Annotations to be added to ApplicationSet controller pdb

2556

annotations: {}

2557

# -- Number of pods that are available after eviction as number or percentage (eg.: 50%)

2558

# @default -- `""` (defaults to 0 if not specified)

2559

minAvailable: ""

2560

# -- Number of pods that are unavailable after eviction as number or percentage (eg.: 50%).

2561

## Has higher precedence over `applicationSet.pdb.minAvailable`

2562

maxUnavailable: ""

2563

​

2564

## ApplicationSet controller image

2565

image:

2566

# -- Repository to use for the ApplicationSet controller

2567

# @default -- `""` (defaults to global.image.repository)

2568

repository: ""

2569

# -- Tag to use for the ApplicationSet controller

2570

# @default -- `""` (defaults to global.image.tag)

2571

tag: ""

2572

# -- Image pull policy for the ApplicationSet controller

2573

# @default -- `""` (defaults to global.image.imagePullPolicy)

2574

imagePullPolicy: ""

2575

​

2576

# -- If defined, uses a Secret to pull an image from a private Docker registry or repository.

2577

# @default -- `[]` (defaults to global.imagePullSecrets)

2578

imagePullSecrets: []

2579

​

2580

# -- ApplicationSet controller command line flags

2581

extraArgs: []

2582

​

2583

# -- Environment variables to pass to the ApplicationSet controller

2584

extraEnv: []

2585

# - name: "MY\_VAR"

2586

# value: "value"

2587

​

2588

# -- envFrom to pass to the ApplicationSet controller

2589

# @default -- `[]` (See [values.yaml])

2590

extraEnvFrom: []

2591

# - configMapRef:

2592

# name: config-map-name

2593

# - secretRef:

2594

# name: secret-name

2595

​

2596

# -- Additional containers to be added to the ApplicationSet controller pod

2597

## Note: Supports use of custom Helm templates

2598

extraContainers: []

2599

​

2600

# -- Init containers to add to the ApplicationSet controller pod

2601

## Note: Supports use of custom Helm templates

2602

initContainers: []

2603

​

2604

# -- List of extra mounts to add (normally used with extraVolumes)

2605

extraVolumeMounts: []

2606

​

2607

# -- List of extra volumes to add

2608

extraVolumes: []

2609

​

2610

## Metrics service configuration

2611

metrics:

2612

# -- Deploy metrics service

2613

enabled: false

2614

service:

2615

# -- Metrics service type

2616

type: ClusterIP

2617

# -- Metrics service clusterIP. `None` makes a "headless service" (no virtual IP)

2618

clusterIP: ""

2619

# -- Metrics service annotations

2620

annotations: {}

2621

# -- Metrics service labels

2622

labels: {}

2623

# -- Metrics service port

2624

servicePort: 8080

2625

# -- Metrics service port name

2626

portName: http-metrics

2627

serviceMonitor:

2628

# -- Enable a prometheus ServiceMonitor

2629

enabled: false

2630

# -- Prometheus ServiceMonitor interval

2631

interval: 30s

2632

# -- Prometheus ServiceMonitor scrapeTimeout. If empty, Prometheus uses the global scrape timeout unless it is less than the target's scrape interval value in which the latter is used.

2633

scrapeTimeout: ""

2634

# -- Prometheus [RelabelConfigs] to apply to samples before scraping

2635

relabelings: []

2636

# -- Prometheus [MetricRelabelConfigs] to apply to samples before ingestion

2637

metricRelabelings: []

2638

# -- Prometheus ServiceMonitor selector

2639

selector: {}

2640

# prometheus: kube-prometheus

2641

​

2642

# -- Prometheus ServiceMonitor scheme

2643

scheme: ""

2644

# -- Prometheus ServiceMonitor tlsConfig

2645

tlsConfig: {}

2646

# -- Prometheus ServiceMonitor namespace

2647

namespace: "" # monitoring

2648

# -- Prometheus ServiceMonitor labels

2649

additionalLabels: {}

2650

# -- Prometheus ServiceMonitor annotations

2651

annotations: {}

2652

​

2653

## ApplicationSet service configuration

2654

service:

2655

# -- ApplicationSet service annotations

2656

annotations: {}

2657

# -- ApplicationSet service labels

2658

labels: {}

2659

# -- ApplicationSet service type

2660

type: ClusterIP

2661

# -- ApplicationSet service port

2662

port: 7000

2663

# -- ApplicationSet service port name

2664

portName: http-webhook

2665

​

2666

serviceAccount:

2667

# -- Create ApplicationSet controller service account

2668

create: true

2669

# -- ApplicationSet controller service account name

2670

name: argocd-applicationset-controller

2671

# -- Annotations applied to created service account

2672

annotations: {}

2673

# -- Labels applied to created service account

2674

labels: {}

2675

# -- Automount API credentials for the Service Account

2676

automountServiceAccountToken: true

2677

​

2678

# -- Annotations to be added to ApplicationSet controller Deployment

2679

deploymentAnnotations: {}

2680

​

2681

# -- Annotations for the ApplicationSet controller pods

2682

podAnnotations: {}

2683

​

2684

# -- Labels for the ApplicationSet controller pods

2685

podLabels: {}

2686

​

2687

# -- Resource limits and requests for the ApplicationSet controller pods.

2688

resources: {}

2689

# limits:

2690

# cpu: 100m

2691

# memory: 128Mi

2692

# requests:

2693

# cpu: 100m

2694

# memory: 128Mi

2695

​

2696

# ApplicationSet controller container ports

2697

containerPorts:

2698

# -- Metrics container port

2699

metrics: 8080

2700

# -- Probe container port

2701

probe: 8081

2702

# -- Webhook container port

2703

webhook: 7000

2704

​

2705

# -- [DNS configuration]

2706

dnsConfig: {}

2707

# -- Alternative DNS policy for ApplicationSet controller pods

2708

dnsPolicy: "ClusterFirst"

2709

​

2710

# -- ApplicationSet controller container-level security context

2711

# @default -- See [values.yaml]

2712

containerSecurityContext:

2713

runAsNonRoot: true

2714

readOnlyRootFilesystem: true

2715

allowPrivilegeEscalation: false

2716

seccompProfile:

2717

type: RuntimeDefault

2718

capabilities:

2719

drop:

2720

- ALL

2721

​

2722

## Probes for ApplicationSet controller (optional)

2723

## Ref: https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readiness-probes/

2724

readinessProbe:

2725

# -- Enable Kubernetes liveness probe for ApplicationSet controller

2726

enabled: false

2727

# -- Number of seconds after the container has started before [probe] is initiated

2728

initialDelaySeconds: 10

2729

# -- How often (in seconds) to perform the [probe]

2730

periodSeconds: 10

2731

# -- Number of seconds after which the [probe] times out

2732

timeoutSeconds: 1

2733

# -- Minimum consecutive successes for the [probe] to be considered successful after having failed

2734

successThreshold: 1

2735

# -- Minimum consecutive failures for the [probe] to be considered failed after having succeeded

2736

failureThreshold: 3

2737

​

2738

livenessProbe:

2739

# -- Enable Kubernetes liveness probe for ApplicationSet controller

2740

enabled: false

2741

# -- Number of seconds after the container has started before [probe] is initiated

2742

initialDelaySeconds: 10

2743

# -- How often (in seconds) to perform the [probe]

2744

periodSeconds: 10

2745

# -- Number of seconds after which the [probe] times out

2746

timeoutSeconds: 1

2747

# -- Minimum consecutive successes for the [probe] to be considered successful after having failed

2748

successThreshold: 1

2749

# -- Minimum consecutive failures for the [probe] to be considered failed after having succeeded

2750

failureThreshold: 3

2751

​

2752

# -- terminationGracePeriodSeconds for container lifecycle hook

2753

terminationGracePeriodSeconds: 30

2754

​

2755

# -- [Node selector]

2756

# @default -- `{}` (defaults to global.nodeSelector)

2757

nodeSelector: {}

2758

​

2759

# -- [Tolerations] for use with node taints

2760

# @default -- `[]` (defaults to global.tolerations)

2761

tolerations: []

2762

​

2763

# -- Assign custom [affinity] rules

2764

# @default -- `{}` (defaults to global.affinity preset)

2765

affinity: {}

2766

​

2767

# -- Assign custom [TopologySpreadConstraints] rules to the ApplicationSet controller

2768

# @default -- `[]` (defaults to global.topologySpreadConstraints)

2769

## Ref: https://kubernetes.io/docs/concepts/workloads/pods/pod-topology-spread-constraints/

2770

## If labelSelector is left out, it will default to the labelSelector configuration of the deployment

2771

topologySpreadConstraints: []

2772

# - maxSkew: 1

2773

# topologyKey: topology.kubernetes.io/zone

2774

# whenUnsatisfiable: DoNotSchedule

2775

​

2776

# -- Deployment strategy to be added to the ApplicationSet controller Deployment

2777

deploymentStrategy: {}

2778

# type: RollingUpdate

2779

# rollingUpdate:

2780

# maxSurge: 25%

2781

# maxUnavailable: 25%

2782

​

2783

# -- Priority class for the ApplicationSet controller pods

2784

# @default -- `""` (defaults to global.priorityClassName)

2785

priorityClassName: ""

2786

​

2787

# TLS certificate configuration via cert-manager

2788

## Ref: https://argo-cd.readthedocs.io/en/stable/operator-manual/tls/#tls-configuration

2789

certificate:

2790

# -- Deploy a Certificate resource (requires cert-manager)

2791

enabled: false

2792

# -- The name of the Secret that will be automatically created and managed by this Certificate resource

2793

secretName: argocd-applicationset-controller-tls

2794

# -- Certificate primary domain (commonName)

2795

domain: argocd.example.com

2796

# -- Certificate Subject Alternate Names (SANs)

2797

additionalHosts: []

2798

# -- The requested 'duration' (i.e. lifetime) of the certificate.

2799

# @default -- `""` (defaults to 2160h = 90d if not specified)

2800

## Ref: https://cert-manager.io/docs/usage/certificate/#renewal

2801

duration: ""

2802

# -- How long before the expiry a certificate should be renewed.

2803

# @default -- `""` (defaults to 360h = 15d if not specified)

2804

## Ref: https://cert-manager.io/docs/usage/certificate/#renewal

2805

renewBefore: ""

2806

# Certificate issuer

2807

## Ref: https://cert-manager.io/docs/concepts/issuer

2808

issuer:

2809

# -- Certificate issuer group. Set if using an external issuer. Eg. `cert-manager.io`

2810

group: ""

2811

# -- Certificate issuer kind. Either `Issuer` or `ClusterIssuer`

2812

kind: ""

2813

# -- Certificate issuer name. Eg. `letsencrypt`

2814

name: ""

2815

# Private key of the certificate

2816

privateKey:

2817

# -- Rotation policy of private key when certificate is re-issued. Either: `Never` or `Always`

2818

rotationPolicy: Never

2819

# -- The private key cryptography standards (PKCS) encoding for private key. Either: `PCKS1` or `PKCS8`

2820

encoding: PKCS1

2821

# -- Algorithm used to generate certificate private key. One of: `RSA`, `Ed25519` or `ECDSA`

2822

algorithm: RSA

2823

# -- Key bit size of the private key. If algorithm is set to `Ed25519`, size is ignored.

2824

size: 2048

2825

# -- Annotations to be applied to the ApplicationSet Certificate

2826

annotations: {}

2827

​

2828

## Ingress for the Git Generator webhook

2829

## Ref: https://argocd-applicationset.readthedocs.io/en/master/Generators-Git/#webhook-configuration)

2830

ingress:

2831

# -- Enable an ingress resource for ApplicationSet webhook

2832

enabled: false

2833

# -- Additional ingress labels

2834

labels: {}

2835

# -- Additional ingress annotations

2836

annotations: {}

2837

​

2838

# -- Defines which ingress ApplicationSet controller will implement the resource

2839

ingressClassName: ""

2840

​

2841

# -- Argo CD ApplicationSet hostname

2842

## NOTE: Hostname must be provided if Ingress is enabled

2843

hostname: argocd.example.com

2844

​

2845

# -- List of ingress paths

2846

path: /api/webhook

2847

​

2848

# -- Ingress path type. One of `Exact`, `Prefix` or `ImplementationSpecific`

2849

pathType: Prefix

2850

​

2851

# -- Enable TLS configuration for the hostname defined at `applicationSet.webhook.ingress.hostname`

2852

## TLS certificate will be retrieved from a TLS secret with name:`argocd-applicationset-controller-tls`

2853

tls: false

2854

​

2855

# -- The list of additional hostnames to be covered by ingress record

2856

# @default -- `[]` (See [values.yaml])

2857

extraHosts: []

2858

# - name: argocd.example.com

2859

# path: /

2860

​

2861

# -- Additional ingress paths

2862

# @default -- `[]` (See [values.yaml])

2863

extraPaths: []

2864

# - path: /\*

2865

# pathType: Prefix

2866

# backend:

2867

# service:

2868

# name: ssl-redirect

2869

# port:

2870

# name: use-annotation

2871

​

2872

# -- Additional ingress rules

2873

# @default -- `[]` (See [values.yaml])

2874

extraRules: []

2875

# - host: example.example.com

2876

# http:

2877

# path: /

2878

# backend:

2879

# service:

2880

# name: example-svc

2881

# port:

2882

# name: http

2883

​

2884

# -- Additional ingress TLS configuration

2885

# @default -- `[]` (See [values.yaml])

2886

extraTls: []

2887

# - secretName: argocd-applicationset-tls

2888

# hosts:

2889

# - argocd-applicationset.example.com

2890

​

2891

## Notifications controller

2892

notifications:

2893

# -- Enable notifications controller

2894

enabled: true

2895

​

2896

# -- Notifications controller name string

2897

name: notifications-controller

2898

​

2899

# -- Argo CD dashboard url; used in place of {{.context.argocdUrl}} in templates

2900

argocdUrl:

2901

​

2902

## Notifications controller Pod Disruption Budget

2903

## Ref: https://kubernetes.io/docs/tasks/run-application/configure-pdb/

2904

pdb:

2905

# -- Deploy a [PodDisruptionBudget] for the notifications controller

2906

enabled: false

2907

# -- Labels to be added to notifications controller pdb

2908

labels: {}

2909

# -- Annotations to be added to notifications controller pdb

2910

annotations: {}

2911

# -- Number of pods that are available after eviction as number or percentage (eg.: 50%)

2912

# @default -- `""` (defaults to 0 if not specified)

2913

minAvailable: ""

2914

# -- Number of pods that are unavailable after eviction as number or percentage (eg.: 50%).

2915

## Has higher precedence over `notifications.pdb.minAvailable`

2916

maxUnavailable: ""

2917

​

2918

## Notifications controller image

2919

image:

2920

# -- Repository to use for the notifications controller

2921

# @default -- `""` (defaults to global.image.repository)

2922

repository: ""

2923

# -- Tag to use for the notifications controller

2924

# @default -- `""` (defaults to global.image.tag)

2925

tag: ""

2926

# -- Image pull policy for the notifications controller

2927

# @default -- `""` (defaults to global.image.imagePullPolicy)

2928

imagePullPolicy: ""

2929

​

2930

# -- Secrets with credentials to pull images from a private registry

2931

# @default -- `[]` (defaults to global.imagePullSecrets)

2932

imagePullSecrets: []

2933

​

2934

# -- Notifications controller log format. Either `text` or `json`

2935

# @default -- `""` (defaults to global.logging.format)

2936

logFormat: ""

2937

# -- Notifications controller log level. One of: `debug`, `info`, `warn`, `error`

2938

# @default -- `""` (defaults to global.logging.level)

2939

logLevel: ""

2940

​

2941

# -- Extra arguments to provide to the notifications controller

2942

extraArgs: []

2943

​

2944

# -- Additional container environment variables

2945

extraEnv: []

2946

​

2947

# -- envFrom to pass to the notifications controller

2948

# @default -- `[]` (See [values.yaml])

2949

extraEnvFrom: []

2950

# - configMapRef:

2951

# name: config-map-name

2952

# - secretRef:

2953

# name: secret-name

2954

​

2955

# -- Additional containers to be added to the notifications controller pod

2956

## Note: Supports use of custom Helm templates

2957

extraContainers: []

2958

​

2959

# -- Init containers to add to the notifications controller pod

2960

## Note: Supports use of custom Helm templates

2961

initContainers: []

2962

​

2963

# -- List of extra mounts to add (normally used with extraVolumes)

2964

extraVolumeMounts: []

2965

​

2966

# -- List of extra volumes to add

2967

extraVolumes: []

2968

​

2969

# -- Define user-defined context

2970

## For more information: https://argo-cd.readthedocs.io/en/stable/operator-manual/notifications/templates/#defining-user-defined-context

2971

context: {}

2972

# region: east

2973

# environmentName: staging

2974

​

2975

secret:

2976

# -- Whether helm chart creates notifications controller secret

2977

create: true

2978

​

2979

# -- key:value pairs of annotations to be added to the secret

2980

annotations: {}

2981

​

2982

# -- key:value pairs of labels to be added to the secret

2983

labels: {}

2984

​

2985

# -- Generic key:value pairs to be inserted into the secret

2986

## Can be used for templates, notification services etc. Some examples given below.

2987

## For more information: https://argo-cd.readthedocs.io/en/stable/operator-manual/notifications/services/overview/

2988

items: {}

2989

# slack-token:

2990

# # For more information: https://argo-cd.readthedocs.io/en/stable/operator-manual/notifications/services/slack/

2991

​

2992

# grafana-apiKey:

2993

# # For more information: https://argo-cd.readthedocs.io/en/stable/operator-manual/notifications/services/grafana/

2994

​

2995

# webhooks-github-token:

2996

​

2997

# email-username:

2998

# email-password:

2999

# For more information: https://argo-cd.readthedocs.io/en/stable/operator-manual/notifications/services/email/

3000

​

3001

metrics:

3002

# -- Enables prometheus metrics server

3003

enabled: false

3004

# -- Metrics port

3005

port: 9001

3006

service:

3007

# -- Metrics service type

3008

type: ClusterIP

3009

# -- Metrics service clusterIP. `None` makes a "headless service" (no virtual IP)

3010

clusterIP: ""

3011

# -- Metrics service annotations

3012

annotations: {}

3013

# -- Metrics service labels

3014

labels: {}

3015

# -- Metrics service port name

3016

portName: http-metrics

3017

serviceMonitor:

3018

# -- Enable a prometheus ServiceMonitor

3019

enabled: false

3020

# -- Prometheus ServiceMonitor selector

3021

selector: {}

3022

# prometheus: kube-prometheus

3023

# -- Prometheus ServiceMonitor labels

3024

additionalLabels: {}

3025

# -- Prometheus ServiceMonitor annotations

3026

annotations: {}

3027

# namespace: monitoring

3028

# interval: 30s

3029

# scrapeTimeout: 10s

3030

# -- Prometheus ServiceMonitor scheme

3031

scheme: ""

3032

# -- Prometheus ServiceMonitor tlsConfig

3033

tlsConfig: {}

3034

# -- Prometheus [RelabelConfigs] to apply to samples before scraping

3035

relabelings: []

3036

# -- Prometheus [MetricRelabelConfigs] to apply to samples before ingestion

3037

metricRelabelings: []

3038

​

3039

# -- Configures notification services such as slack, email or custom webhook

3040

# @default -- See [values.yaml]

3041

## For more information: https://argo-cd.readthedocs.io/en/stable/operator-manual/notifications/services/overview/

3042

notifiers: {}

3043

# service.slack: |

3044

# token: $slack-token

3045

​

3046

# -- Annotations to be applied to the notifications controller Deployment

3047

deploymentAnnotations: {}

3048

​

3049

# -- Annotations to be applied to the notifications controller Pods

3050

podAnnotations: {}

3051

​

3052

# -- Labels to be applied to the notifications controller Pods

3053

podLabels: {}

3054

​

3055

# -- Resource limits and requests for the notifications controller

3056

resources: {}

3057

# limits:

3058

# cpu: 100m

3059

# memory: 128Mi

3060

# requests:

3061

# cpu: 100m

3062

# memory: 128Mi

3063

​

3064

# Notification controller container ports

3065

containerPorts:

3066

# -- Metrics container port

3067

metrics: 9001

3068

​

3069

# -- [DNS configuration]

3070

dnsConfig: {}

3071

# -- Alternative DNS policy for notifications controller Pods

3072

dnsPolicy: "ClusterFirst"

3073

​

3074

# -- Notification controller container-level security Context

3075

# @default -- See [values.yaml]

3076

containerSecurityContext:

3077

runAsNonRoot: true

3078

readOnlyRootFilesystem: true

3079

allowPrivilegeEscalation: false

3080

seccompProfile:

3081

type: RuntimeDefault

3082

capabilities:

3083

drop:

3084

- ALL

3085

​

3086

# -- terminationGracePeriodSeconds for container lifecycle hook

3087

terminationGracePeriodSeconds: 30

3088

​

3089

# -- [Node selector]

3090

# @default -- `{}` (defaults to global.nodeSelector)

3091

nodeSelector: {}

3092

​

3093

# -- [Tolerations] for use with node taints

3094

# @default -- `[]` (defaults to global.tolerations)

3095

tolerations: []

3096

​

3097

# -- Assign custom [affinity] rules

3098

# @default -- `{}` (defaults to global.affinity preset)

3099

affinity: {}

3100

​

3101

# -- Assign custom [TopologySpreadConstraints] rules to the application controller

3102

# @default -- `[]` (defaults to global.topologySpreadConstraints)

3103

## Ref: https://kubernetes.io/docs/concepts/workloads/pods/pod-topology-spread-constraints/

3104

## If labelSelector is left out, it will default to the labelSelector configuration of the deployment

3105

topologySpreadConstraints: []

3106

# - maxSkew: 1

3107

# topologyKey: topology.kubernetes.io/zone

3108

# whenUnsatisfiable: DoNotSchedule

3109

​

3110

# -- Deployment strategy to be added to the notifications controller Deployment

3111

deploymentStrategy:

3112

type: Recreate

3113

​

3114

# -- Priority class for the notifications controller pods

3115

# @default -- `""` (defaults to global.priorityClassName)

3116

priorityClassName: ""

3117

​

3118

serviceAccount:

3119

# -- Create notifications controller service account

3120

create: true

3121

# -- Notification controller service account name

3122

name: argocd-notifications-controller

3123

# -- Annotations applied to created service account

3124

annotations: {}

3125

# -- Labels applied to created service account

3126

labels: {}

3127

# -- Automount API credentials for the Service Account

3128

automountServiceAccountToken: true

3129

​

3130

cm:

3131

# -- Whether helm chart creates notifications controller config map

3132

create: true

3133

​

3134

## Enable this and set the rules: to whatever custom rules you want for the Cluster Role resource.

3135

## Defaults to off

3136

clusterRoleRules:

3137

# -- List of custom rules for the notifications controller's ClusterRole resource

3138

rules: []

3139

​

3140

# -- Contains centrally managed global application subscriptions

3141

## For more information: https://argo-cd.readthedocs.io/en/stable/operator-manual/notifications/subscriptions/

3142

subscriptions: []

3143

# # subscription for on-sync-status-unknown trigger notifications

3144

# - recipients:

3145

# - slack:test2

3146

# - email:[email protected]

3147

# triggers:

3148

# - on-sync-status-unknown

3149

# # subscription restricted to applications with matching labels only

3150

# - recipients:

3151

# - slack:test3

3152

# selector: test=true

3153

# triggers:

3154

# - on-sync-status-unknown

3155

​

3156

# -- The notification template is used to generate the notification content

3157

## For more information: https://argo-cd.readthedocs.io/en/stable/operator-manual/notifications/templates/

3158

templates: {}

3159

# template.app-deployed: |

3160

# email:

3161

# subject: New version of an application {{.app.metadata.name}} is up and running.

3162

# message: |

3163

# {{if eq .serviceType "slack"}}:white\_check\_mark:{{end}} Application {{.app.metadata.name}} is now running new version of deployments manifests.

3164

# slack:

3165

# attachments: |

3166

# [{

3167

# "title": "{{ .app.metadata.name}}",

3168

# "title\_link":"{{.context.argocdUrl}}/applications/{{.app.metadata.name}}",

3169

# "color": "#18be52",

3170

# "fields": [

3171

# {

3172

# "title": "Sync Status",

3173

# "value": "{{.app.status.sync.status}}",

3174

# "short": true

3175

# },

3176

# {

3177

# "title": "Repository",

3178

# "value": "{{.app.spec.source.repoURL}}",

3179

# "short": true

3180

# },

3181

# {

3182

# "title": "Revision",

3183

# "value": "{{.app.status.sync.revision}}",

3184

# "short": true

3185

# }

3186

# {{range $index, $c := .app.status.conditions}}

3187

# {{if not $index}},{{end}}

3188

# {{if $index}},{{end}}

3189

# {

3190

# "title": "{{$c.type}}",

3191

# "value": "{{$c.message}}",

3192

# "short": true

3193

# }

3194

# {{end}}

3195

# ]

3196

# }]

3197

# template.app-health-degraded: |

3198

# email:

3199

# subject: Application {{.app.metadata.name}} has degraded.

3200

# message: |

3201

# {{if eq .serviceType "slack"}}:exclamation:{{end}} Application {{.app.metadata.name}} has degraded.

3202

# Application details: {{.context.argocdUrl}}/applications/{{.app.metadata.name}}.

3203

# slack:

3204

# attachments: |-

3205

# [{

3206

# "title": "{{ .app.metadata.name}}",

3207

# "title\_link": "{{.context.argocdUrl}}/applications/{{.app.metadata.name}}",

3208

# "color": "#f4c030",

3209

# "fields": [

3210

# {

3211

# "title": "Sync Status",

3212

# "value": "{{.app.status.sync.status}}",

3213

# "short": true

3214

# },

3215

# {

3216

# "title": "Repository",

3217

# "value": "{{.app.spec.source.repoURL}}",

3218

# "short": true

3219

# }

3220

# {{range $index, $c := .app.status.conditions}}

3221

# {{if not $index}},{{end}}

3222

# {{if $index}},{{end}}

3223

# {

3224

# "title": "{{$c.type}}",

3225

# "value": "{{$c.message}}",

3226

# "short": true

3227

# }

3228

# {{end}}

3229

# ]

3230

# }]

3231

# template.app-sync-failed: |

3232

# email:

3233

# subject: Failed to sync application {{.app.metadata.name}}.

3234

# message: |

3235

# {{if eq .serviceType "slack"}}:exclamation:{{end}} The sync operation of application {{.app.metadata.name}} has failed at {{.app.status.operationState.finishedAt}} with the following error: {{.app.status.operationState.message}}

3236

# Sync operation details are available at: {{.context.argocdUrl}}/applications/{{.app.metadata.name}}?operation=true .

3237

# slack:

3238

# attachments: |-

3239

# [{

3240

# "title": "{{ .app.metadata.name}}",

3241

# "title\_link":"{{.context.argocdUrl}}/applications/{{.app.metadata.name}}",

3242

# "color": "#E96D76",

3243

# "fields": [

3244

# {

3245

# "title": "Sync Status",

3246

# "value": "{{.app.status.sync.status}}",

3247

# "short": true

3248

# },

3249

# {

3250

# "title": "Repository",

3251

# "value": "{{.app.spec.source.repoURL}}",

3252

# "short": true

3253

# }

3254

# {{range $index, $c := .app.status.conditions}}

3255

# {{if not $index}},{{end}}

3256

# {{if $index}},{{end}}

3257

# {

3258

# "title": "{{$c.type}}",

3259

# "value": "{{$c.message}}",

3260

# "short": true

3261

# }

3262

# {{end}}

3263

# ]

3264

# }]

3265

# template.app-sync-running: |

3266

# email:

3267

# subject: Start syncing application {{.app.metadata.name}}.

3268

# message: |

3269

# The sync operation of application {{.app.metadata.name}} has started at {{.app.status.operationState.startedAt}}.

3270

# Sync operation details are available at: {{.context.argocdUrl}}/applications/{{.app.metadata.name}}?operation=true .

3271

# slack:

3272

# attachments: |-

3273

# [{

3274

# "title": "{{ .app.metadata.name}}",

3275

# "title\_link":"{{.context.argocdUrl}}/applications/{{.app.metadata.name}}",

3276

# "color": "#0DADEA",

3277

# "fields": [

3278

# {

3279

# "title": "Sync Status",

3280

# "value": "{{.app.status.sync.status}}",

3281

# "short": true

3282

# },

3283

# {

3284

# "title": "Repository",

3285

# "value": "{{.app.spec.source.repoURL}}",

3286

# "short": true

3287

# }

3288

# {{range $index, $c := .app.status.conditions}}

3289

# {{if not $index}},{{end}}

3290

# {{if $index}},{{end}}

3291

# {

3292

# "title": "{{$c.type}}",

3293

# "value": "{{$c.message}}",

3294

# "short": true

3295

# }

3296

# {{end}}

3297

# ]

3298

# }]

3299

# template.app-sync-status-unknown: |

3300

# email:

3301

# subject: Application {{.app.metadata.name}} sync status is 'Unknown'

3302

# message: |

3303

# {{if eq .serviceType "slack"}}:exclamation:{{end}} Application {{.app.metadata.name}} sync is 'Unknown'.

3304

# Application details: {{.context.argocdUrl}}/applications/{{.app.metadata.name}}.

3305

# {{if ne .serviceType "slack"}}

3306

# {{range $c := .app.status.conditions}}

3307

# \* {{$c.message}}

3308

# {{end}}

3309

# {{end}}

3310

# slack:

3311

# attachments: |-

3312

# [{

3313

# "title": "{{ .app.metadata.name}}",

3314

# "title\_link":"{{.context.argocdUrl}}/applications/{{.app.metadata.name}}",

3315

# "color": "#E96D76",

3316

# "fields": [

3317

# {

3318

# "title": "Sync Status",

3319

# "value": "{{.app.status.sync.status}}",

3320

# "short": true

3321

# },

3322

# {

3323

# "title": "Repository",

3324

# "value": "{{.app.spec.source.repoURL}}",

3325

# "short": true

3326

# }

3327

# {{range $index, $c := .app.status.conditions}}

3328

# {{if not $index}},{{end}}

3329

# {{if $index}},{{end}}

3330

# {

3331

# "title": "{{$c.type}}",

3332

# "value": "{{$c.message}}",

3333

# "short": true

3334

# }

3335

# {{end}}

3336

# ]

3337

# }]

3338

# template.app-sync-succeeded: |

3339

# email:

3340

# subject: Application {{.app.metadata.name}} has been successfully synced.

3341

# message: |

3342

# {{if eq .serviceType "slack"}}:white\_check\_mark:{{end}} Application {{.app.metadata.name}} has been successfully synced at {{.app.status.operationState.finishedAt}}.

3343

# Sync operation details are available at: {{.context.argocdUrl}}/applications/{{.app.metadata.name}}?operation=true .

3344

# slack:

3345

# attachments: |-

3346

# [{

3347

# "title": "{{ .app.metadata.name}}",

3348

# "title\_link":"{{.context.argocdUrl}}/applications/{{.app.metadata.name}}",

3349

# "color": "#18be52",

3350

# "fields": [

3351

# {

3352

# "title": "Sync Status",

3353

# "value": "{{.app.status.sync.status}}",

3354

# "short": true

3355

# },

3356

# {

3357

# "title": "Repository",

3358

# "value": "{{.app.spec.source.repoURL}}",

3359

# "short": true

3360

# }

3361

# {{range $index, $c := .app.status.conditions}}

3362

# {{if not $index}},{{end}}

3363

# {{if $index}},{{end}}

3364

# {

3365

# "title": "{{$c.type}}",

3366

# "value": "{{$c.message}}",

3367

# "short": true

3368

# }

3369

# {{end}}

3370

# ]

3371

# }]

3372

​

3373

# -- The trigger defines the condition when the notification should be sent

3374

## For more information: https://argo-cd.readthedocs.io/en/stable/operator-manual/notifications/triggers/

3375

triggers: {}

3376

# trigger.on-deployed: |

3377

# - description: Application is synced and healthy. Triggered once per commit.

3378

# oncePer: app.status.sync.revision

3379

# send:

3380

# - app-deployed

3381

# when: app.status.operationState.phase in ['Succeeded'] and app.status.health.status == 'Healthy'

3382

# trigger.on-health-degraded: |

3383

# - description: Application has degraded

3384

# send:

3385

# - app-health-degraded

3386

# when: app.status.health.status == 'Degraded'

3387

# trigger.on-sync-failed: |

3388

# - description: Application syncing has failed

3389

# send:

3390

# - app-sync-failed

3391

# when: app.status.operationState.phase in ['Error', 'Failed']

3392

# trigger.on-sync-running: |

3393

# - description: Application is being synced

3394

# send:

3395

# - app-sync-running

3396

# when: app.status.operationState.phase in ['Running']

3397

# trigger.on-sync-status-unknown: |

3398

# - description: Application status is 'Unknown'

3399

# send:

3400

# - app-sync-status-unknown

3401

# when: app.status.sync.status == 'Unknown'

3402

# trigger.on-sync-succeeded: |

3403

# - description: Application syncing has succeeded

3404

# send:

3405

# - app-sync-succeeded

3406

# when: app.status.operationState.phase in ['Succeeded']

3407

#

3408

# For more information: https://argo-cd.readthedocs.io/en/stable/operator-manual/notifications/triggers/#default-triggers

3409

# defaultTriggers: |

3410

# - on-sync-status-unknown

File values trên đã được tùy chỉnh tham số sau để tắt SSL mặc đinh của ArgoCD, vì chúng ta sử dụng cert và domain của công ty

1

configs:

2

params:

3

server.insecure: true

**- Cài đăt ArgoCD với version cụ thể**

1

helm install argo-cd argo/argo-cd --version 6.0.13 --values=argocd-values.yaml --namespace argocd --create-namespace

Ở lệnh trên helm chart version của ArgoCD là 6.0.13 tương ứng với ArgoCD version v2.10.0 và lấy giá trị các tham số từ file argocd-values.yaml, ArgoCD sẽ được cài đặt trên namespace argocd

**- Để tìm version ArgoCD để cài đặt có thể dùng lệnh sau**

1

helm search repo argocd -l

**- Để lấy mật khẩu admin đăng nhập trên giao diện ArgoCD , ta dùng lệnh sau**

1

kubectl -n default get secret argocd-initial-admin-secret -o jsonpath="{.data.password}" | base64 -d

**- Gỡ cài đăt ArgoCD ta dùng lênh sau**

1

helm uninstall argo-cd -n argocd

**- Cuối cùng để có thể truy cập giao diện ArgoCD thông qua domain , ta tiến hành tạo ingress**

1

apiVersion: networking.k8s.io/v1

2

kind: Ingress

3

metadata:

4

annotations:

5

name: argocd-ingress

6

namespace: argocd

7

spec:

8

ingressClassName: nginx

9

rules:

10

- host: argocd.example.com

11

http:

12

paths:

13

- backend:

14

service:

15

name: argo-cd-argocd-server

16

port:

17

number: 80

18

path: /

19

pathType: Prefix

20

tls:

21

- hosts:

22

- argocd.example.com

23

secretName: argocd-tls