



OpenShift Container Platform 4.18

Template APIs

Reference guide for template APIs

OpenShift Container Platform 4.18 Template APIs

Reference guide for template APIs

Legal Notice

Copyright © 2025 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

This document describes the OpenShift Container Platform template API objects and their detailed specifications.

Table of Contents

CHAPTER 1. TEMPLATE APIS	10
1.1. BROKERTEMPLATEINSTANCE [TEMPLATE.OPENSIFT.IO/V1]	10
1.2. PODTEMPLATE [V1]	10
1.3. TEMPLATE [TEMPLATE.OPENSIFT.IO/V1]	10
1.4. TEMPLATEINSTANCE [TEMPLATE.OPENSIFT.IO/V1]	10
CHAPTER 2. BROKERTEMPLATEINSTANCE [TEMPLATE.OPENSIFT.IO/V1]	11
2.1. SPECIFICATION	11
2.1.1. .spec	12
2.2. API ENDPOINTS	12
2.2.1. /apis/template.openshift.io/v1/brokertemplateinstances	13
2.2.2. /apis/template.openshift.io/v1/watch/brokertemplateinstances	15
2.2.3. /apis/template.openshift.io/v1/brokertemplateinstances/{name}	15
2.2.4. /apis/template.openshift.io/v1/watch/brokertemplateinstances/{name}	18
CHAPTER 3. PODTEMPLATE [V1]	20
3.1. SPECIFICATION	20
3.1.1. .template	20
3.1.2. .template.spec	21
3.1.3. .template.spec.affinity	32
3.1.4. .template.spec.affinity.nodeAffinity	32
3.1.5. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution	33
3.1.6. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[]	33
3.1.7. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference	34
3.1.8. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matc	35
3.1.9. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matc	35
3.1.10. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.mat	36
3.1.11. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution[].preference.matc	36
3.1.12. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution	37
3.1.13. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerm	37
3.1.14. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerm	37
3.1.15. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerm	38
3.1.16. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerm	38
3.1.17. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerm	39
3.1.18. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringExecution.nodeSelectorTerm	39
3.1.19. .template.spec.affinity.podAffinity	40
3.1.20. .template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution	42
3.1.21. .template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[]	42
3.1.22. .template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinityTerm	43
3.1.23. .template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution	45
3.1.24. .template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution[]	45
3.1.25. .template.spec.affinity.podAntiAffinity	48

3.1.26. .template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution	49
3.1.27. .template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[]	49
3.1.28. .template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution[].podAffinity	50
3.1.29. .template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution	52
3.1.30. .template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution[]	52
3.1.31. .template.spec.containers	55
3.1.32. .template.spec.containers[]	55
3.1.33. .template.spec.containers[].env	63
3.1.34. .template.spec.containers[].env[]	63
3.1.35. .template.spec.containers[].env[].valueFrom	64
3.1.36. .template.spec.containers[].env[].valueFrom.configMapKeyRef	64
3.1.37. .template.spec.containers[].env[].valueFrom.fieldRef	65
3.1.38. .template.spec.containers[].env[].valueFrom.resourceFieldRef	65
3.1.39. .template.spec.containers[].env[].valueFrom.secretKeyRef	66
3.1.40. .template.spec.containers[].envFrom	66
3.1.41. .template.spec.containers[].envFrom[]	67
3.1.42. .template.spec.containers[].envFrom[].configMapRef	67
3.1.43. .template.spec.containers[].envFrom[].secretRef	68
3.1.44. .template.spec.containers[].lifecycle	68
3.1.45. .template.spec.containers[].lifecycle.postStart	69
3.1.46. .template.spec.containers[].lifecycle.postStart.exec	69
3.1.47. .template.spec.containers[].lifecycle.postStart.httpGet	70
3.1.48. .template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders	71
3.1.49. .template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders[]	71
3.1.50. .template.spec.containers[].lifecycle.postStart.sleep	71
3.1.51. .template.spec.containers[].lifecycle.postStart.tcpSocket	72
3.1.52. .template.spec.containers[].lifecycle.preStop	72
3.1.53. .template.spec.containers[].lifecycle.preStop.exec	73
3.1.54. .template.spec.containers[].lifecycle.preStop.httpGet	73
3.1.55. .template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders	74
3.1.56. .template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders[]	74
3.1.57. .template.spec.containers[].lifecycle.preStop.sleep	75
3.1.58. .template.spec.containers[].lifecycle.preStop.tcpSocket	75
3.1.59. .template.spec.containers[].livenessProbe	76
3.1.60. .template.spec.containers[].livenessProbe.exec	77
3.1.61. .template.spec.containers[].livenessProbe.grpc	78
3.1.62. .template.spec.containers[].livenessProbe.httpGet	78
3.1.63. .template.spec.containers[].livenessProbe.httpGet.httpHeaders	79
3.1.64. .template.spec.containers[].livenessProbe.httpGet.httpHeaders[]	79
3.1.65. .template.spec.containers[].livenessProbe.tcpSocket	80
3.1.66. .template.spec.containers[].ports	80
3.1.67. .template.spec.containers[].ports[]	80
3.1.68. .template.spec.containers[].readinessProbe	81
3.1.69. .template.spec.containers[].readinessProbe.exec	83
3.1.70. .template.spec.containers[].readinessProbe.grpc	84
3.1.71. .template.spec.containers[].readinessProbe.httpGet	84
3.1.72. .template.spec.containers[].readinessProbe.httpGet.httpHeaders	85
3.1.73. .template.spec.containers[].readinessProbe.httpGet.httpHeaders[]	85
3.1.74. .template.spec.containers[].readinessProbe.tcpSocket	86
3.1.75. .template.spec.containers[].resizePolicy	86
3.1.76. .template.spec.containers[].resizePolicy[]	86
3.1.77. .template.spec.containers[].resources	87

3.1.78. .template.spec.containers[].resources.claims	88
3.1.79. .template.spec.containers[].resources.claims[]	88
3.1.80. .template.spec.containers[].securityContext	89
3.1.81. .template.spec.containers[].securityContext.appArmorProfile	91
3.1.82. .template.spec.containers[].securityContext.capabilities	92
3.1.83. .template.spec.containers[].securityContext.seLinuxOptions	92
3.1.84. .template.spec.containers[].securityContext.seccompProfile	93
3.1.85. .template.spec.containers[].securityContext.windowsOptions	94
3.1.86. .template.spec.containers[].startupProbe	95
3.1.87. .template.spec.containers[].startupProbe.exec	97
3.1.88. .template.spec.containers[].startupProbe.grpc	97
3.1.89. .template.spec.containers[].startupProbe.httpGet	98
3.1.90. .template.spec.containers[].startupProbe.httpGet.httpHeaders	99
3.1.91. .template.spec.containers[].startupProbe.httpGet.httpHeaders[]	99
3.1.92. .template.spec.containers[].startupProbe.tcpSocket	99
3.1.93. .template.spec.containers[].volumeDevices	100
3.1.94. .template.spec.containers[].volumeDevices[]	100
3.1.95. .template.spec.containers[].volumeMounts	100
3.1.96. .template.spec.containers[].volumeMounts[]	101
3.1.97. .template.spec.dnsConfig	103
3.1.98. .template.spec.dnsConfig.options	104
3.1.99. .template.spec.dnsConfig.options[]	104
3.1.100. .template.spec.ephemeralContainers	105
3.1.101. .template.spec.ephemeralContainers[]	105
3.1.102. .template.spec.ephemeralContainers[].env	112
3.1.103. .template.spec.ephemeralContainers[].env[]	112
3.1.104. .template.spec.ephemeralContainers[].env[].valueFrom	113
3.1.105. .template.spec.ephemeralContainers[].env[].valueFrom.configMapKeyRef	113
3.1.106. .template.spec.ephemeralContainers[].env[].valueFrom.fieldRef	114
3.1.107. .template.spec.ephemeralContainers[].env[].valueFrom.resourceFieldRef	114
3.1.108. .template.spec.ephemeralContainers[].env[].valueFrom.secretKeyRef	115
3.1.109. .template.spec.ephemeralContainers[].envFrom	115
3.1.110. .template.spec.ephemeralContainers[].envFrom[]	116
3.1.111. .template.spec.ephemeralContainers[].envFrom[].configMapRef	116
3.1.112. .template.spec.ephemeralContainers[].envFrom[].secretRef	117
3.1.113. .template.spec.ephemeralContainers[].lifecycle	117
3.1.114. .template.spec.ephemeralContainers[].lifecycle.postStart	118
3.1.115. .template.spec.ephemeralContainers[].lifecycle.postStart.exec	118
3.1.116. .template.spec.ephemeralContainers[].lifecycle.postStart.httpGet	119
3.1.117. .template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders	120
3.1.118. .template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeaders[]	120
3.1.119. .template.spec.ephemeralContainers[].lifecycle.postStart.sleep	120
3.1.120. .template.spec.ephemeralContainers[].lifecycle.postStart.tcpSocket	121
3.1.121. .template.spec.ephemeralContainers[].lifecycle.preStop	121
3.1.122. .template.spec.ephemeralContainers[].lifecycle.preStop.exec	122
3.1.123. .template.spec.ephemeralContainers[].lifecycle.preStop.httpGet	122
3.1.124. .template.spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeaders	123
3.1.125. .template.spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeaders[]	123
3.1.126. .template.spec.ephemeralContainers[].lifecycle.preStop.sleep	124
3.1.127. .template.spec.ephemeralContainers[].lifecycle.preStop.tcpSocket	124
3.1.128. .template.spec.ephemeralContainers[].livenessProbe	125
3.1.129. .template.spec.ephemeralContainers[].livenessProbe.exec	126
3.1.130. .template.spec.ephemeralContainers[].livenessProbe.grpc	127

3.1.131. .template.spec.ephemeralContainers[].livenessProbe.httpGet	127
3.1.132. .template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders	128
3.1.133. .template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders[]	128
3.1.134. .template.spec.ephemeralContainers[].livenessProbe.tcpSocket	129
3.1.135. .template.spec.ephemeralContainers[].ports	129
3.1.136. .template.spec.ephemeralContainers[].ports[]	129
3.1.137. .template.spec.ephemeralContainers[].readinessProbe	130
3.1.138. .template.spec.ephemeralContainers[].readinessProbe.exec	132
3.1.139. .template.spec.ephemeralContainers[].readinessProbe.grpc	133
3.1.140. .template.spec.ephemeralContainers[].readinessProbe.httpGet	133
3.1.141. .template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders	134
3.1.142. .template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders[]	134
3.1.143. .template.spec.ephemeralContainers[].readinessProbe.tcpSocket	135
3.1.144. .template.spec.ephemeralContainers[].resizePolicy	135
3.1.145. .template.spec.ephemeralContainers[].resizePolicy[]	135
3.1.146. .template.spec.ephemeralContainers[].resources	136
3.1.147. .template.spec.ephemeralContainers[].resources.claims	137
3.1.148. .template.spec.ephemeralContainers[].resources.claims[]	137
3.1.149. .template.spec.ephemeralContainers[].securityContext	138
3.1.150. .template.spec.ephemeralContainers[].securityContext.appArmorProfile	140
3.1.151. .template.spec.ephemeralContainers[].securityContext.capabilities	141
3.1.152. .template.spec.ephemeralContainers[].securityContext.seLinuxOptions	141
3.1.153. .template.spec.ephemeralContainers[].securityContext.seccompProfile	142
3.1.154. .template.spec.ephemeralContainers[].securityContext.windowsOptions	143
3.1.155. .template.spec.ephemeralContainers[].startupProbe	144
3.1.156. .template.spec.ephemeralContainers[].startupProbe.exec	146
3.1.157. .template.spec.ephemeralContainers[].startupProbe.grpc	146
3.1.158. .template.spec.ephemeralContainers[].startupProbe.httpGet	147
3.1.159. .template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders	148
3.1.160. .template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders[]	148
3.1.161. .template.spec.ephemeralContainers[].startupProbe.tcpSocket	148
3.1.162. .template.spec.ephemeralContainers[].volumeDevices	149
3.1.163. .template.spec.ephemeralContainers[].volumeDevices[]	149
3.1.164. .template.spec.ephemeralContainers[].volumeMounts	149
3.1.165. .template.spec.ephemeralContainers[].volumeMounts[]	150
3.1.166. .template.spec.hostAliases	152
3.1.167. .template.spec.hostAliases[]	153
3.1.168. .template.spec.imagePullSecrets	153
3.1.169. .template.spec.imagePullSecrets[]	153
3.1.170. .template.spec.initContainers	154
3.1.171. .template.spec.initContainers[]	154
3.1.172. .template.spec.initContainers[].env	162
3.1.173. .template.spec.initContainers[].env[]	162
3.1.174. .template.spec.initContainers[].env[].valueFrom	163
3.1.175. .template.spec.initContainers[].env[].valueFrom.configMapKeyRef	163
3.1.176. .template.spec.initContainers[].env[].valueFrom.fieldRef	164
3.1.177. .template.spec.initContainers[].env[].valueFrom.resourceFieldRef	164
3.1.178. .template.spec.initContainers[].env[].valueFrom.secretKeyRef	165
3.1.179. .template.spec.initContainers[].envFrom	165
3.1.180. .template.spec.initContainers[].envFrom[]	166
3.1.181. .template.spec.initContainers[].envFrom[].configMapRef	166
3.1.182. .template.spec.initContainers[].envFrom[].secretRef	167
3.1.183. .template.spec.initContainers[].lifecycle	167

3.1.184. .template.spec.initContainers[].lifecycle.postStart	168
3.1.185. .template.spec.initContainers[].lifecycle.postStart.exec	168
3.1.186. .template.spec.initContainers[].lifecycle.postStart.httpGet	169
3.1.187. .template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders	170
3.1.188. .template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders[]	170
3.1.189. .template.spec.initContainers[].lifecycle.postStart.sleep	170
3.1.190. .template.spec.initContainers[].lifecycle.postStart.tcpSocket	171
3.1.191. .template.spec.initContainers[].lifecycle.preStop	171
3.1.192. .template.spec.initContainers[].lifecycle.preStop.exec	172
3.1.193. .template.spec.initContainers[].lifecycle.preStop.httpGet	172
3.1.194. .template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders	173
3.1.195. .template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders[]	173
3.1.196. .template.spec.initContainers[].lifecycle.preStop.sleep	174
3.1.197. .template.spec.initContainers[].lifecycle.preStop.tcpSocket	174
3.1.198. .template.spec.initContainers[].livenessProbe	175
3.1.199. .template.spec.initContainers[].livenessProbe.exec	176
3.1.200. .template.spec.initContainers[].livenessProbe.grpc	177
3.1.201. .template.spec.initContainers[].livenessProbe.httpGet	177
3.1.202. .template.spec.initContainers[].livenessProbe.httpGet.httpHeaders	178
3.1.203. .template.spec.initContainers[].livenessProbe.httpGet.httpHeaders[]	178
3.1.204. .template.spec.initContainers[].livenessProbe.tcpSocket	179
3.1.205. .template.spec.initContainers[].ports	179
3.1.206. .template.spec.initContainers[].ports[]	179
3.1.207. .template.spec.initContainers[].readinessProbe	180
3.1.208. .template.spec.initContainers[].readinessProbe.exec	182
3.1.209. .template.spec.initContainers[].readinessProbe.grpc	183
3.1.210. .template.spec.initContainers[].readinessProbe.httpGet	183
3.1.211. .template.spec.initContainers[].readinessProbe.httpGet.httpHeaders	184
3.1.212. .template.spec.initContainers[].readinessProbe.httpGet.httpHeaders[]	184
3.1.213. .template.spec.initContainers[].readinessProbe.tcpSocket	185
3.1.214. .template.spec.initContainers[].resizePolicy	185
3.1.215. .template.spec.initContainers[].resizePolicy[]	185
3.1.216. .template.spec.initContainers[].resources	186
3.1.217. .template.spec.initContainers[].resources.claims	187
3.1.218. .template.spec.initContainers[].resources.claims[]	187
3.1.219. .template.spec.initContainers[].securityContext	188
3.1.220. .template.spec.initContainers[].securityContext.appArmorProfile	190
3.1.221. .template.spec.initContainers[].securityContext.capabilities	191
3.1.222. .template.spec.initContainers[].securityContext.seLinuxOptions	191
3.1.223. .template.spec.initContainers[].securityContext.seccompProfile	192
3.1.224. .template.spec.initContainers[].securityContext.windowsOptions	193
3.1.225. .template.spec.initContainers[].startupProbe	194
3.1.226. .template.spec.initContainers[].startupProbe.exec	196
3.1.227. .template.spec.initContainers[].startupProbe.grpc	196
3.1.228. .template.spec.initContainers[].startupProbe.httpGet	197
3.1.229. .template.spec.initContainers[].startupProbe.httpGet.httpHeaders	198
3.1.230. .template.spec.initContainers[].startupProbe.httpGet.httpHeaders[]	198
3.1.231. .template.spec.initContainers[].startupProbe.tcpSocket	198
3.1.232. .template.spec.initContainers[].volumeDevices	199
3.1.233. .template.spec.initContainers[].volumeDevices[]	199
3.1.234. .template.spec.initContainers[].volumeMounts	199
3.1.235. .template.spec.initContainers[].volumeMounts[]	200
3.1.236. .template.spec.os	202

3.1.237. .template.spec.readinessGates	203
3.1.238. .template.spec.readinessGates[]	203
3.1.239. .template.spec.resourceClaims	204
3.1.240. .template.spec.resourceClaims[]	204
3.1.241. .template.spec.schedulingGates	205
3.1.242. .template.spec.schedulingGates[]	205
3.1.243. .template.spec.securityContext	206
3.1.244. .template.spec.securityContext.appArmorProfile	210
3.1.245. .template.spec.securityContext.seLinuxOptions	211
3.1.246. .template.spec.securityContext.seccompProfile	211
3.1.247. .template.spec.securityContext.sysctls	212
3.1.248. .template.spec.securityContext.sysctls[]	213
3.1.249. .template.spec.securityContext.windowsOptions	213
3.1.250. .template.spec.tolerations	214
3.1.251. .template.spec.tolerations[]	214
3.1.252. .template.spec.topologySpreadConstraints	216
3.1.253. .template.spec.topologySpreadConstraints[]	216
3.1.254. .template.spec.volumes	222
3.1.255. .template.spec.volumes[]	222
3.1.256. .template.spec.volumes[].awsElasticBlockStore	227
3.1.257. .template.spec.volumes[].azureDisk	228
3.1.258. .template.spec.volumes[].azureFile	229
3.1.259. .template.spec.volumes[].cephfs	230
3.1.260. .template.spec.volumes[].cephfs.secretRef	231
3.1.261. .template.spec.volumes[].cinder	231
3.1.262. .template.spec.volumes[].cinder.secretRef	232
3.1.263. .template.spec.volumes[].configMap	233
3.1.264. .template.spec.volumes[].configMap.items	234
3.1.265. .template.spec.volumes[].configMap.items[]	234
3.1.266. .template.spec.volumes[].csi	235
3.1.267. .template.spec.volumes[].csi.nodePublishSecretRef	236
3.1.268. .template.spec.volumes[].downwardAPI	237
3.1.269. .template.spec.volumes[].downwardAPI.items	237
3.1.270. .template.spec.volumes[].downwardAPI.items[]	237
3.1.271. .template.spec.volumes[].downwardAPI.items[].fieldRef	238
3.1.272. .template.spec.volumes[].downwardAPI.items[].resourceFieldRef	239
3.1.273. .template.spec.volumes[].emptyDir	239
3.1.274. .template.spec.volumes[].ephemeral	240
3.1.275. .template.spec.volumes[].ephemeral.volumeClaimTemplate	240
3.1.276. .template.spec.volumes[].ephemeral.volumeClaimTemplate.spec	241
3.1.277. .template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSource	245
3.1.278. .template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSourceRef	245
3.1.279. .template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources	247
3.1.280. .template.spec.volumes[].fc	247
3.1.281. .template.spec.volumes[].flexVolume	248
3.1.282. .template.spec.volumes[].flexVolume.secretRef	249
3.1.283. .template.spec.volumes[].flocker	249
3.1.284. .template.spec.volumes[].gcePersistentDisk	250
3.1.285. .template.spec.volumes[].gitRepo	251
3.1.286. .template.spec.volumes[].glusterfs	252
3.1.287. .template.spec.volumes[].hostPath	253
3.1.288. .template.spec.volumes[].image	254
3.1.289. .template.spec.volumes[].iscsi	255

3.1.290. .template.spec.volumes[].iscsi.secretRef	257
3.1.291. .template.spec.volumes[].nfs	257
3.1.292. .template.spec.volumes[].persistentVolumeClaim	258
3.1.293. .template.spec.volumes[].photonPersistentDisk	259
3.1.294. .template.spec.volumes[].portworxVolume	259
3.1.295. .template.spec.volumes[].projected	260
3.1.296. .template.spec.volumes[].projected.sources	261
3.1.297. .template.spec.volumes[].projected.sources[]	261
3.1.298. .template.spec.volumes[].projected.sources[].clusterTrustBundle	262
3.1.299. .template.spec.volumes[].projected.sources[].configMap	263
3.1.300. .template.spec.volumes[].projected.sources[].configMap.items	264
3.1.301. .template.spec.volumes[].projected.sources[].configMap.items[]	264
3.1.302. .template.spec.volumes[].projected.sources[].downwardAPI	265
3.1.303. .template.spec.volumes[].projected.sources[].downwardAPI.items	266
3.1.304. .template.spec.volumes[].projected.sources[].downwardAPI.items[]	266
3.1.305. .template.spec.volumes[].projected.sources[].downwardAPI.items[].fieldRef	266
3.1.306. .template.spec.volumes[].projected.sources[].downwardAPI.items[].resourceFieldRef	267
3.1.307. .template.spec.volumes[].projected.sources[].secret	267
3.1.308. .template.spec.volumes[].projected.sources[].secret.items	268
3.1.309. .template.spec.volumes[].projected.sources[].secret.items[]	268
3.1.310. .template.spec.volumes[].projected.sources[].serviceAccountToken	269
3.1.311. .template.spec.volumes[].quobyte	270
3.1.312. .template.spec.volumes[].rbd	271
3.1.313. .template.spec.volumes[].rbd.secretRef	273
3.1.314. .template.spec.volumes[].scaleIO	273
3.1.315. .template.spec.volumes[].scaleIO.secretRef	274
3.1.316. .template.spec.volumes[].secret	275
3.1.317. .template.spec.volumes[].secret.items	276
3.1.318. .template.spec.volumes[].secret.items[]	276
3.1.319. .template.spec.volumes[].storageos	277
3.1.320. .template.spec.volumes[].storageos.secretRef	278
3.1.321. .template.spec.volumes[].vsphereVolume	278
3.2. API ENDPOINTS	279
3.2.1. /api/v1/podtemplates	280
3.2.2. /api/v1/watch/podtemplates	280
3.2.3. /api/v1/namespaces/{namespace}/podtemplates	281
3.2.4. /api/v1/watch/namespaces/{namespace}/podtemplates	282
3.2.5. /api/v1/namespaces/{namespace}/podtemplates/{name}	283
3.2.6. /api/v1/watch/namespaces/{namespace}/podtemplates/{name}	286
CHAPTER 4. TEMPLATE [TEMPLATE.OPENSIFT.IO/V1]	287
4.1. SPECIFICATION	287
4.1.1. .parameters	288
4.1.2. .parameters[]	289
4.2. API ENDPOINTS	290
4.2.1. /apis/template.openshift.io/v1/templates	291
4.2.2. /apis/template.openshift.io/v1/watch/templates	291
4.2.3. /apis/template.openshift.io/v1/namespaces/{namespace}/templates	291
4.2.4. /apis/template.openshift.io/v1/watch/namespaces/{namespace}/templates	293
4.2.5. /apis/template.openshift.io/v1/namespaces/{namespace}/templates/{name}	294
4.2.6. /apis/template.openshift.io/v1/namespaces/{namespace}/processedtemplates	297
4.2.7. /apis/template.openshift.io/v1/watch/namespaces/{namespace}/templates/{name}	298

CHAPTER 5. TEMPLATEINSTANCE [TEMPLATE.OPENSIFT.IO/V1]	299
5.1. SPECIFICATION	299
5.1.1. .spec	300
5.1.2. .spec.requester	300
5.1.3. .spec.requester.extra	301
5.1.4. .spec.template	301
5.1.5. .spec.template.parameters	303
5.1.6. .spec.template.parameters[]	303
5.1.7. .status	305
5.1.8. .status.conditions	305
5.1.9. .status.conditions[]	305
5.1.10. .status.objects	306
5.1.11. .status.objects[]	306
5.2. API ENDPOINTS	306
5.2.1. /apis/template.openshift.io/v1/templateinstances	307
5.2.2. /apis/template.openshift.io/v1/watch/templateinstances	308
5.2.3. /apis/template.openshift.io/v1/namespaces/{namespace}/templateinstances	308
5.2.4. /apis/template.openshift.io/v1/watch/namespaces/{namespace}/templateinstances	310
5.2.5. /apis/template.openshift.io/v1/namespaces/{namespace}/templateinstances/{name}	310
5.2.6. /apis/template.openshift.io/v1/watch/namespaces/{namespace}/templateinstances/{name}	313
5.2.7. /apis/template.openshift.io/v1/namespaces/{namespace}/templateinstances/{name}/status	314

CHAPTER 1. TEMPLATE APIS

1.1. BROKERTEMPLATEINSTANCE [TEMPLATE.OPENSIFT.IO/V1]

Description

BrokerTemplateInstance holds the service broker-related state associated with a TemplateInstance. BrokerTemplateInstance is part of an experimental API.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.2. PODTEMPLATE [V1]

Description

PodTemplate describes a template for creating copies of a predefined pod.

Type

object

1.3. TEMPLATE [TEMPLATE.OPENSIFT.IO/V1]

Description

Template contains the inputs needed to produce a Config.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

1.4. TEMPLATEINSTANCE [TEMPLATE.OPENSIFT.IO/V1]

Description

TemplateInstance requests and records the instantiation of a Template. TemplateInstance is part of an experimental API.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

CHAPTER 2. BROKERTEMPLATEINSTANCE [TEMPLATE.OPENSIFT.IO/V1]

Description

BrokerTemplateInstance holds the service broker-related state associated with a TemplateInstance. BrokerTemplateInstance is part of an experimental API.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **spec**

2.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta_v2	metadata is the standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

Property	Type	Description
spec	object	BrokerTemplateInstanceSpec describes the state of a BrokerTemplateInstance.

2.1.1. .spec

Description

BrokerTemplateInstanceSpec describes the state of a BrokerTemplateInstance.

Type

object

Required

- **templateInstance**
- **secret**

Property	Type	Description
bindingIDs	array (string)	bindingids is a list of 'binding_id's provided during successive bind calls to the template service broker.
secret	ObjectReference	secret is a reference to a Secret object residing in a namespace, containing the necessary template parameters.
templateInstance	ObjectReference	templateinstance is a reference to a TemplateInstance object residing in a namespace.

2.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/template.openshift.io/v1/brokertemplateinstances**
 - **DELETE**: delete collection of BrokerTemplateInstance
 - **GET**: list or watch objects of kind BrokerTemplateInstance
 - **POST**: create a BrokerTemplateInstance
- **/apis/template.openshift.io/v1/watch/brokertemplateinstances**

- **GET**: watch individual changes to a list of BrokerTemplateInstance. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/template.openshift.io/v1/brokertemplateinstances/{name}**
 - **DELETE**: delete a BrokerTemplateInstance
 - **GET**: read the specified BrokerTemplateInstance
 - **PATCH**: partially update the specified BrokerTemplateInstance
 - **PUT**: replace the specified BrokerTemplateInstance
- **/apis/template.openshift.io/v1/watch/brokertemplateinstances/{name}**
 - **GET**: watch changes to an object of kind BrokerTemplateInstance. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

2.2.1. /apis/template.openshift.io/v1/brokertemplateinstances

HTTP method

DELETE

Description

delete collection of BrokerTemplateInstance

Table 2.1. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 2.2. HTTP responses

HTTP code	Response body
200 - OK	Status_v9 schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list or watch objects of kind BrokerTemplateInstance

Table 2.3. HTTP responses

HTTP code	Response body
200 - OK	BrokerTemplateInstanceList schema
401 - Unauthorized	Empty

HTTP method**POST****Description**

create a BrokerTemplateInstance

Table 2.4. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 2.5. Body parameters

Parameter	Type	Description
body	BrokerTemplateInstance schema	

Table 2.6. HTTP responses

HTTP code	Reponse body
200 - OK	BrokerTemplateInstance schema
201 - Created	BrokerTemplateInstance schema
202 - Accepted	BrokerTemplateInstance schema
401 - Unauthorized	Empty

2.2.2. /apis/template.openshift.io/v1/watch/brokertemplateinstances

HTTP method

GET

Description

watch individual changes to a list of BrokerTemplateInstance. deprecated: use the 'watch' parameter with a list operation instead.

Table 2.7. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

2.2.3. /apis/template.openshift.io/v1/brokertemplateinstances/{name}

Table 2.8. Global path parameters

Parameter	Type	Description
name	string	name of the BrokerTemplateInstance

HTTP method

DELETE

Description

delete a BrokerTemplateInstance

Table 2.9. Query parameters

Parameter	Type	Description
-----------	------	-------------

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 2.10. HTTP responses

HTTP code	Reponse body
200 - OK	Status_v9 schema
202 - Accepted	Status_v9 schema
401 - Unauthorized	Empty

HTTP method**GET****Description**

read the specified BrokerTemplateInstance

Table 2.11. HTTP responses

HTTP code	Reponse body
200 - OK	BrokerTemplateInstance schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update the specified BrokerTemplateInstance

Table 2.12. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 2.13. HTTP responses

HTTP code	Response body
200 - OK	BrokerTemplateInstance schema
201 - Created	BrokerTemplateInstance schema
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace the specified BrokerTemplateInstance

Table 2.14. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: <ul style="list-style-type: none"> - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 2.15. Body parameters

Parameter	Type	Description
body	BrokerTemplateInstance schema	

Table 2.16. HTTP responses

HTTP code	Response body
200 - OK	BrokerTemplateInstance schema
201 - Created	BrokerTemplateInstance schema
401 - Unauthorized	Empty

2.2.4. /apis/template.openshift.io/v1/watch/brokertemplateinstances/{name}

Table 2.17. Global path parameters

Parameter	Type	Description
name	string	name of the BrokerTemplateInstance

HTTP method

GET

Description

watch changes to an object of kind BrokerTemplateInstance. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 2.18. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

CHAPTER 3. PODTEMPLATE [V1]

Description

PodTemplate describes a template for creating copies of a predefined pod.

Type

object

3.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
template	object	PodTemplateSpec describes the data a pod should have when created from a template

3.1.1. .template

Description

PodTemplateSpec describes the data a pod should have when created from a template

Type

object

Property	Type	Description
metadata	ObjectMeta	Standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
spec	object	PodSpec is a description of a pod.

3.1.2. .template.spec**Description**

PodSpec is a description of a pod.

Type

object

Required

- **containers**

Property	Type	Description
activeDeadlineSeconds	integer	Optional duration in seconds the pod may be active on the node relative to StartTime before the system will actively try to mark it failed and kill associated containers. Value must be a positive integer.
affinity	object	Affinity is a group of affinity scheduling rules.
automountServiceAccountToken	boolean	AutomountServiceAccountToken indicates whether a service account token should be automatically mounted.
containers	array	List of containers belonging to the pod. Containers cannot currently be added or removed. There must be at least one container in a Pod. Cannot be updated.

Property	Type	Description
containers[]	object	A single application container that you want to run within a pod.
dnsConfig	object	PodDNSConfig defines the DNS parameters of a pod in addition to those generated from DNSPolicy.
dnsPolicy	string	<p>Set DNS policy for the pod. Defaults to "ClusterFirst". Valid values are 'ClusterFirstWithHostNet', 'ClusterFirst', 'Default' or 'None'. DNS parameters given in DNSConfig will be merged with the policy selected with DNSPolicy. To have DNS options set along with hostNetwork, you have to specify DNS policy explicitly to 'ClusterFirstWithHostNet'.</p> <p>Possible enum values: - "ClusterFirst" indicates that the pod should use cluster DNS first unless hostNetwork is true, if it is available, then fall back on the default (as determined by kubelet) DNS settings. - "ClusterFirstWithHostNet" indicates that the pod should use cluster DNS first, if it is available, then fall back on the default (as determined by kubelet) DNS settings. - "Default" indicates that the pod should use the default (as determined by kubelet) DNS settings. - "None" indicates that the pod should use empty DNS settings. DNS parameters such as nameservers and search paths should be defined via DNSConfig.</p>
enableServiceLinks	boolean	EnableServiceLinks indicates whether information about services should be injected into pod's environment variables, matching the syntax of Docker links. Optional: Defaults to true.

Property	Type	Description
ephemeralContainers	array	List of ephemeral containers run in this pod. Ephemeral containers may be run in an existing pod to perform user-initiated actions such as debugging. This list cannot be specified when creating a pod, and it cannot be modified by updating the pod spec. In order to add an ephemeral container to an existing pod, use the pod's <code>ephemeralcontainers</code> subresource.
ephemeralContainers[]	object	<p>An EphemeralContainer is a temporary container that you may add to an existing Pod for user-initiated activities such as debugging. Ephemeral containers have no resource or scheduling guarantees, and they will not be restarted when they exit or when a Pod is removed or restarted. The kubelet may evict a Pod if an ephemeral container causes the Pod to exceed its resource allocation.</p> <p>To add an ephemeral container, use the <code>ephemeralcontainers</code> subresource of an existing Pod. Ephemeral containers may not be removed or restarted.</p>
hostAliases	array	HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified.
hostAliases[]	object	HostAlias holds the mapping between IP and hostnames that will be injected as an entry in the pod's hosts file.
hostIPC	boolean	Use the host's ipc namespace. Optional: Default to false.

Property	Type	Description
hostNetwork	boolean	Host networking requested for this pod. Use the host's network namespace. If this option is set, the ports that will be used must be specified. Default to false.
hostPID	boolean	Use the host's pid namespace. Optional: Default to false.
hostUsers	boolean	Use the host's user namespace. Optional: Default to true. If set to true or not present, the pod will be run in the host user namespace, useful for when the pod needs a feature only available to the host user namespace, such as loading a kernel module with CAP_SYS_MODULE. When set to false, a new users is created for the pod. Setting false is useful for mitigating container breakout vulnerabilities even allowing users to run their containers as root without actually having root privileges on the host. This field is alpha-level and is only honored by servers that enable the UserNamespacesSupport feature.
hostname	string	Specifies the hostname of the Pod. If not specified, the pod's hostname will be set to a system-defined value.
imagePullSecrets	array	ImagePullSecrets is an optional list of references to secrets in the same namespace to use for pulling any of the images used by this PodSpec. If specified, these secrets will be passed to individual puller implementations for them to use. More info: https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod
imagePullSecrets[]	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Property	Type	Description
initContainers	array	<p>List of initialization containers belonging to the pod. Init containers are executed in order prior to containers being started. If any init container fails, the pod is considered to have failed and is handled according to its restartPolicy. The name for an init container or normal container must be unique among all containers. Init containers may not have Lifecycle actions, Readiness probes, Liveness probes, or Startup probes. The resourceRequirements of an init container are taken into account during scheduling by finding the highest request/limit for each resource type, and then using the max of that value or the sum of the normal containers. Limits are applied to init containers in a similar fashion. Init containers cannot currently be added or removed. Cannot be updated.</p> <p>More info: https://kubernetes.io/docs/concepts/workloads/pods/init-containers/</p>
initContainers[]	object	A single application container that you want to run within a pod.
nodeName	string	<p>nodeName indicates in which node this pod is scheduled. If empty, this pod is a candidate for scheduling by the scheduler defined in schedulerName. Once this field is set, the kubelet for this node becomes responsible for the lifecycle of this pod. This field should not be used to express a desire for the pod to be scheduled on a specific node.</p> <p>https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/#nodename</p>

Property	Type	Description
nodeSelector	object (string)	<p>NodeSelector is a selector which must be true for the pod to fit on a node. Selector which must match a node's labels for the pod to be scheduled on that node.</p> <p>More info: https://kubernetes.io/docs/concepts/configuration/assign-pod-node/</p>
os	object	PodOS defines the OS parameters of a pod.
overhead	object (Quantity)	<p>Overhead represents the resource overhead associated with running a pod for a given RuntimeClass. This field will be autopopulated at admission time by the RuntimeClass admission controller. If the RuntimeClass admission controller is enabled, overhead must not be set in Pod create requests. The RuntimeClass admission controller will reject Pod create requests which have the overhead already set. If RuntimeClass is configured and selected in the PodSpec, Overhead will be set to the value defined in the corresponding RuntimeClass, otherwise it will remain unset and treated as zero. More info: https://git.k8s.io/enhancements/keps/sig-node/688-pod-overhead/README.md</p>
preemptionPolicy	string	<p>PreemptionPolicy is the Policy for preempting pods with lower priority. One of Never, PreemptLowerPriority. Defaults to PreemptLowerPriority if unset.</p> <p>Possible enum values: - "Never" means that pod never preempts other pods with lower priority. - "PreemptLowerPriority" means that pod can preempt other pods with lower priority.</p>

Property	Type	Description
priority	integer	The priority value. Various system components use this field to find the priority of the pod. When Priority Admission Controller is enabled, it prevents users from setting this field. The admission controller populates this field from PriorityClassName. The higher the value, the higher the priority.
priorityClassName	string	If specified, indicates the pod's priority. "system-node-critical" and "system-cluster-critical" are two special keywords which indicate the highest priorities with the former being the highest priority. Any other name must be defined by creating a PriorityClass object with that name. If not specified, the pod priority will be default or zero if there is no default.
readinessGates	array	If specified, all readiness gates will be evaluated for pod readiness. A pod is ready when all its containers are ready AND all conditions specified in the readiness gates have status equal to "True" More info: https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates
readinessGates[]	object	PodReadinessGate contains the reference to a pod condition

Property	Type	Description
resourceClaims	array	<p>ResourceClaims defines which ResourceClaims must be allocated and reserved before the Pod is allowed to start. The resources will be made available to those containers which consume them by name.</p> <p>This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.</p> <p>This field is immutable.</p>
resourceClaims[]	object	<p>PodResourceClaim references exactly one ResourceClaim, either directly or by naming a ResourceClaimTemplate which is then turned into a ResourceClaim for the pod.</p> <p>It adds a name to it that uniquely identifies the ResourceClaim inside the Pod. Containers that need access to the ResourceClaim reference it with this name.</p>
restartPolicy	string	<p>Restart policy for all containers within the pod. One of Always, OnFailure, Never. In some contexts, only a subset of those values may be permitted. Default to Always. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle/#restart-policy</p> <p>Possible enum values: - "Always" - "Never" - "OnFailure"</p>

Property	Type	Description
runtimeClassName	string	RuntimeClassName refers to a RuntimeClass object in the node.k8s.io group, which should be used to run this pod. If no RuntimeClass resource matches the named class, the pod will not be run. If unset or empty, the "legacy" RuntimeClass will be used, which is an implicit class with an empty definition that uses the default runtime handler. More info: https://git.k8s.io/enhancements/keps/sig-node/585-runtime-class
schedulerName	string	If specified, the pod will be dispatched by specified scheduler. If not specified, the pod will be dispatched by default scheduler.
schedulingGates	array	<p>SchedulingGates is an opaque list of values that if specified will block scheduling the pod. If schedulingGates is not empty, the pod will stay in the SchedulingGated state and the scheduler will not attempt to schedule the pod.</p> <p>SchedulingGates can only be set at pod creation time, and be removed only afterwards.</p>
schedulingGates[]	object	PodSchedulingGate is associated to a Pod to guard its scheduling.
securityContext	object	PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.

Property	Type	Description
serviceAccount	string	DeprecatedServiceAccount is a deprecated alias for ServiceAccountName. Deprecated: Use serviceAccountName instead.
serviceAccountName	string	ServiceAccountName is the name of the ServiceAccount to use to run this pod. More info: https://kubernetes.io/docs/tasks/configure-pod-container/configure-service-account/
setHostnameAsFQDN	boolean	If true the pod's hostname will be configured as the pod's FQDN, rather than the leaf name (the default). In Linux containers, this means setting the FQDN in the hostname field of the kernel (the nodename field of struct utsname). In Windows containers, this means setting the registry value of hostname for the registry key HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters to FQDN. If a pod does not have FQDN, this has no effect. Default to false.
shareProcessNamespace	boolean	Share a single process namespace between all of the containers in a pod. When this is set containers will be able to view and signal processes from other containers in the same pod, and the first process in each container will not be assigned PID 1. HostPID and ShareProcessNamespace cannot both be set. Optional: Default to false.
subdomain	string	If specified, the fully qualified Pod hostname will be "<hostname>.<subdomain>.<pod namespace>.svc.<cluster domain>". If not specified, the pod will not have a domainname at all.

Property	Type	Description
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully. May be decreased in delete request. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). If this value is nil, the default grace period will be used instead. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. Defaults to 30 seconds.
tolerations	array	If specified, the pod's tolerations.
tolerations[]	object	The pod this Tolerant is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.
topologySpreadConstraints	array	TopologySpreadConstraints describes how a group of pods ought to spread across topology domains. Scheduler will schedule pods in a way which abides by the constraints. All topologySpreadConstraints are ANDed.
topologySpreadConstraints[]	object	TopologySpreadConstraint specifies how to spread matching pods among the given topology.
volumes	array	List of volumes that can be mounted by containers belonging to the pod. More info: https://kubernetes.io/docs/concepts/storage/volumes

Property	Type	Description
volumes[]	object	Volume represents a named volume in a pod that may be accessed by any container in the pod.

3.1.3. .template.spec.affinity

Description

Affinity is a group of affinity scheduling rules.

Type

object

Property	Type	Description
nodeAffinity	object	Node affinity is a group of node affinity scheduling rules.
podAffinity	object	Pod affinity is a group of inter pod affinity scheduling rules.
podAntiAffinity	object	Pod anti affinity is a group of inter pod anti affinity scheduling rules.

3.1.4. .template.spec.affinity.nodeAffinity

Description

Node affinity is a group of node affinity scheduling rules.

Type

object

Property	Type	Description
----------	------	-------------

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.
preferredDuringSchedulingIgnoredDuringExecution[]	object	An empty preferred scheduling term matches all objects with implicit weight 0 (i.e. it's a no-op). A null preferred scheduling term matches no objects (i.e. is also a no-op).
requiredDuringSchedulingIgnoredDuringExecution	object	A node selector represents the union of the results of one or more label queries over a set of nodes; that is, it represents the OR of the selectors represented by the node selector terms.

3.1.5. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node matches the corresponding matchExpressions; the node(s) with the highest sum are the most preferred.

Type

array

3.1.6. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

An empty preferred scheduling term matches all objects with implicit weight 0 (i.e. it's a no-op). A null preferred scheduling term matches no objects (i.e. is also a no-op).

Type

object

Required

- **weight**
- **preference**

Property	Type	Description
preference	object	A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.
weight	integer	Weight associated with matching the corresponding nodeSelectorTerm, in the range 1-100.

3.1.7. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution**Description**

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

Type

object

Property	Type	Description
matchExpressions	array	A list of node selector requirements by node's labels.
matchExpressions[]	object	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchFields	array	A list of node selector requirements by node's fields.

Property	Type	Description
matchFields[]	object	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

3.1.8. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

A list of node selector requirements by node's labels.

Type

array

3.1.9. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

- **key**
- **operator**

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt. Possible enum values: - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"

Property	Type	Description
values	array (string)	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

3.1.10. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringE

Description

A list of node selector requirements by node's fields.

Type

array

3.1.11. .template.spec.affinity.nodeAffinity.preferredDuringSchedulingIgnoredDuringE

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

- **key**
- **operator**

Property	Type	Description
key	string	The label key that the selector applies to.
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist. Gt, and Lt. Possible enum values: - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"

Property	Type	Description
values	array (string)	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

3.1.12. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringEx

Description

A node selector represents the union of the results of one or more label queries over a set of nodes; that is, it represents the OR of the selectors represented by the node selector terms.

Type

object

Required

- **nodeSelectorTerms**

Property	Type	Description
nodeSelectorTerms	array	Required. A list of node selector terms. The terms are ORed.
nodeSelectorTerms[]	object	A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

3.1.13. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringEx

Description

Required. A list of node selector terms. The terms are ORed.

Type

array

3.1.14. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringEx

Description

A null or empty node selector term matches no objects. The requirements of them are ANDed. The TopologySelectorTerm type implements a subset of the NodeSelectorTerm.

Type

object

Property	Type	Description
matchExpressions	array	A list of node selector requirements by node's labels.
matchExpressions[]	object	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.
matchFields	array	A list of node selector requirements by node's fields.
matchFields[]	object	A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

3.1.15. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringEx

Description

A list of node selector requirements by node's labels.

Type

array

3.1.16. .template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringEx

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

- **key**
- **operator**

Property	Type	Description
key	string	The label key that the selector applies to.

Property	Type	Description
operator	string	Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt. Possible enum values: - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"
values	array (string)	An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.

3.1.17. `.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringEx`

Description

A list of node selector requirements by node's fields.

Type

array

3.1.18. `.template.spec.affinity.nodeAffinity.requiredDuringSchedulingIgnoredDuringEx`

Description

A node selector requirement is a selector that contains values, a key, and an operator that relates the key and values.

Type

object

Required

- **key**
- **operator**

Property	Type	Description
key	string	The label key that the selector applies to.

Property	Type	Description
operator	string	<p>Represents a key's relationship to a set of values. Valid operators are In, NotIn, Exists, DoesNotExist, Gt, and Lt.</p> <p>Possible enum values: - "DoesNotExist" - "Exists" - "Gt" - "In" - "Lt" - "NotIn"</p>
values	array (string)	<p>An array of string values. If the operator is In or NotIn, the values array must be non-empty. If the operator is Exists or DoesNotExist, the values array must be empty. If the operator is Gt or Lt, the values array must have a single element, which will be interpreted as an integer. This array is replaced during a strategic merge patch.</p>

3.1.19. .template.spec.affinity.podAffinity

Description

Pod affinity is a group of inter pod affinity scheduling rules.

Type

object

Property	Type	Description
----------	------	-------------

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.
preferredDuringSchedulingIgnoredDuringExecution[]	object	The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)
requiredDuringSchedulingIgnoredDuringExecution	array	If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Property	Type	Description
requiredDuringSchedulingIgnoredDuringExecution[]	object	Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

3.1.20. .template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringEx

Description

The scheduler will prefer to schedule pods to nodes that satisfy the affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

3.1.21. .template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringEx

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

- **weight**
- **podAffinityTerm**

Property	Type	Description
----------	------	-------------

Property	Type	Description
podAffinityTerm	object	Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running
weight	integer	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

3.1.22. .template.spec.affinity.podAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

Type

object

Required

- **topologyKey**

Property	Type	Description
labelSelector	LabelSelector	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Property	Type	Description
matchLabelKeys	array (string)	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
mismatchLabelKeys	array (string)	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key not in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Property	Type	Description
namespaceSelector	LabelSelector	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
namespaces	array (string)	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
topologyKey	string	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

3.1.23. `.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution`

Description

If the affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

3.1.24. `.template.spec.affinity.podAffinity.requiredDuringSchedulingIgnoredDuringExecution`

Description

Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located

is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

Type

object

Required

- **topologyKey**

Property	Type	Description
labelSelector	LabelSelector	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
matchLabelKeys	array (string)	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Property	Type	Description
mismatchLabelKeys	array (string)	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key notin (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
namespaceSelector	LabelSelector	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
namespaces	array (string)	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Property	Type	Description
topologyKey	string	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

3.1.25. .template.spec.affinity.podAntiAffinity

Description

Pod anti affinity is a group of inter pod anti affinity scheduling rules.

Type

object

Property	Type	Description
preferredDuringSchedulingIgnoredDuringExecution	array	The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.
preferredDuringSchedulingIgnoredDuringExecution[]	object	The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Property	Type	Description
requiredDuringSchedulingIgnoredDuringExecution	array	If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.
requiredDuringSchedulingIgnoredDuringExecution[]	object	Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

3.1.26. .template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

The scheduler will prefer to schedule pods to nodes that satisfy the anti-affinity expressions specified by this field, but it may choose a node that violates one or more of the expressions. The node that is most preferred is the one with the greatest sum of weights, i.e. for each node that meets all of the scheduling requirements (resource request, requiredDuringScheduling anti-affinity expressions, etc.), compute a sum by iterating through the elements of this field and adding "weight" to the sum if the node has pods which matches the corresponding podAffinityTerm; the node(s) with the highest sum are the most preferred.

Type

array

3.1.27. .template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution

Description

The weights of all of the matched WeightedPodAffinityTerm fields are added per-node to find the most preferred node(s)

Type

object

Required

- **weight**
- **podAffinityTerm**

Property	Type	Description
podAffinityTerm	object	Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running
weight	integer	weight associated with matching the corresponding podAffinityTerm, in the range 1-100.

3.1.28. .template.spec.affinity.podAntiAffinity.preferredDuringSchedulingIgnoredDuringExecution**Description**

Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

Type**object****Required**

- **topologyKey**

Property	Type	Description
labelSelector	LabelSelector	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.

Property	Type	Description
matchLabelKeys	array (string)	<p>MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).</p>
mismatchLabelKeys	array (string)	<p>MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key not in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).</p>

Property	Type	Description
namespaceSelector	LabelSelector	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
namespaces	array (string)	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".
topologyKey	string	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

3.1.29. .template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

If the anti-affinity requirements specified by this field are not met at scheduling time, the pod will not be scheduled onto the node. If the anti-affinity requirements specified by this field cease to be met at some point during pod execution (e.g. due to a pod label update), the system may or may not try to eventually evict the pod from its node. When there are multiple elements, the lists of nodes corresponding to each podAffinityTerm are intersected, i.e. all terms must be satisfied.

Type

array

3.1.30. .template.spec.affinity.podAntiAffinity.requiredDuringSchedulingIgnoredDuringExecution

Description

Defines a set of pods (namely those matching the labelSelector relative to the given namespace(s)) that this pod should be co-located (affinity) or not co-located (anti-affinity) with, where co-located

is defined as running on a node whose value of the label with key <topologyKey> matches that of any node on which a pod of the set of pods is running

Type

object

Required

- **topologyKey**

Property	Type	Description
labelSelector	LabelSelector	A label query over a set of resources, in this case pods. If it's null, this PodAffinityTerm matches with no Pods.
matchLabelKeys	array (string)	MatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key in (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both matchLabelKeys and labelSelector. Also, matchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).

Property	Type	Description
mismatchLabelKeys	array (string)	MismatchLabelKeys is a set of pod label keys to select which pods will be taken into consideration. The keys are used to lookup values from the incoming pod labels, those key-value labels are merged with labelSelector as key notin (value) to select the group of existing pods which pods will be taken into consideration for the incoming pod's pod (anti) affinity. Keys that don't exist in the incoming pod labels will be ignored. The default value is empty. The same key is forbidden to exist in both mismatchLabelKeys and labelSelector. Also, mismatchLabelKeys cannot be set when labelSelector isn't set. This is a beta field and requires enabling MatchLabelKeysInPodAffinity feature gate (enabled by default).
namespaceSelector	LabelSelector	A label query over the set of namespaces that the term applies to. The term is applied to the union of the namespaces selected by this field and the ones listed in the namespaces field. null selector and null or empty namespaces list means "this pod's namespace". An empty selector ({}) matches all namespaces.
namespaces	array (string)	namespaces specifies a static list of namespace names that the term applies to. The term is applied to the union of the namespaces listed in this field and the ones selected by namespaceSelector. null or empty namespaces list and null namespaceSelector means "this pod's namespace".

Property	Type	Description
topologyKey	string	This pod should be co-located (affinity) or not co-located (anti-affinity) with the pods matching the labelSelector in the specified namespaces, where co-located is defined as running on a node whose value of the label with key topologyKey matches that of any node on which any of the selected pods is running. Empty topologyKey is not allowed.

3.1.31. .template.spec.containers

Description

List of containers belonging to the pod. Containers cannot currently be added or removed. There must be at least one container in a Pod. Cannot be updated.

Type

array

3.1.32. .template.spec.containers[]

Description

A single application container that you want to run within a pod.

Type

object

Required

- **name**

Property	Type	Description
----------	------	-------------

Property	Type	Description
args	array (string)	<p>Arguments to the entrypoint. The container image's CMD is used if this is not provided. Variable references <code>\$(VAR_NAME)</code> are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single <code>\$</code>, which allows for escaping the <code>\$(VAR_NAME)</code> syntax: i.e. "<code>\$(VAR_NAME)</code>" will produce the string literal "<code>\$(VAR_NAME)</code>". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell</p>
command	array (string)	<p>Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references <code>\$(VAR_NAME)</code> are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single <code>\$</code>, which allows for escaping the <code>\$(VAR_NAME)</code> syntax: i.e. "<code>\$(VAR_NAME)</code>" will produce the string literal "<code>\$(VAR_NAME)</code>". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell</p>
env	array	<p>List of environment variables to set in the container. Cannot be updated.</p>

Property	Type	Description
env[]	object	EnvVar represents an environment variable present in a Container.
envFrom	array	List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
envFrom[]	object	EnvFromSource represents the source of a set of ConfigMaps
image	string	Container image name. More info: https://kubernetes.io/docs/concepts/containers/images This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.

Property	Type	Description
imagePullPolicy	string	<p>Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images#updating-images</p> <p>Possible enum values: - "Always" means that kubelet always attempts to pull the latest image. Container will fail if the pull fails. - "IfNotPresent" means that kubelet pulls if the image isn't present on disk. Container will fail if the image isn't present and the pull fails. - "Never" means that kubelet never pulls an image, but only uses a local image. Container will fail if the image isn't present</p>
lifecycle	object	<p>Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.</p>
livenessProbe	object	<p>Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.</p>
name	string	<p>Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL). Cannot be updated.</p>

Property	Type	Description
ports	array	List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See https://github.com/kubernetes/kubernetes/issues/108255 . Cannot be updated.
ports[]	object	ContainerPort represents a network port in a single container.
readinessProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
resizePolicy	array	Resources resize policy for the container.
resizePolicy[]	object	ContainerResizePolicy represents resource resize policy for the container.
resources	object	ResourceRequirements describes the compute resource requirements.

Property	Type	Description
restartPolicy	string	RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers, and the only allowed value is "Always". For non-init containers or when this field is not specified, the restart behavior is defined by the Pod's restart policy and the container type. Setting the RestartPolicy as "Always" for the init container will have the following effect: this init container will be continually restarted on exit until all regular containers have terminated. Once all regular containers have completed, all init containers with restartPolicy "Always" will be shut down. This lifecycle differs from normal init containers and is often referred to as a "sidecar" container. Although this init container still starts in the init container sequence, it does not wait for the container to complete before proceeding to the next init container. Instead, the next init container starts immediately after this init container is started, or after any startupProbe has successfully completed.
securityContext	object	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
startupProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Property	Type	Description
stdin	boolean	Whether this container should allocate a buffer for stdin in the container runtime. If this is not set, reads from stdin in the container will always result in EOF. Default is false.
stdinOnce	boolean	Whether the container runtime should close the stdin channel after it has been opened by a single attach. When stdin is true the stdin stream will remain open across multiple attach sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, and then remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container processes that reads from stdin will never receive an EOF. Default is false
terminationMessagePath	string	Optional: Path at which the file to which the container's termination message will be written is mounted into the container's filesystem. Message written is intended to be brief final status, such as an assertion failure message. Will be truncated by the node if greater than 4096 bytes. The total message length across all containers will be limited to 12kb. Defaults to /dev/termination-log. Cannot be updated.

Property	Type	Description
terminationMessagePolicy	string	<p>Indicate how the termination message should be populated. File will use the contents of terminationMessagePath to populate the container status message on both success and failure. FallbackToLogsOnError will use the last chunk of container log output if the termination message file is empty and the container exited with an error. The log output is limited to 2048 bytes or 80 lines, whichever is smaller. Defaults to File. Cannot be updated.</p> <p>Possible enum values: - "FallbackToLogsOnError" will read the most recent contents of the container logs for the container status message when the container exits with an error and the terminationMessagePath has no contents. - "File" is the default behavior and will set the container status message to the contents of the container's terminationMessagePath when the container exits.</p>
tty	boolean	Whether this container should allocate a TTY for itself, also requires 'stdin' to be true. Default is false.
volumeDevices	array	volumeDevices is the list of block devices to be used by the container.
volumeDevices[]	object	volumeDevice describes a mapping of a raw block device within a container.
volumeMounts	array	Pod volumes to mount into the container's filesystem. Cannot be updated.
volumeMounts[]	object	VolumeMount describes a mounting of a Volume within a container.

Property	Type	Description
workingDir	string	Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image. Cannot be updated.

3.1.33. .template.spec.containers[].env

Description

List of environment variables to set in the container. Cannot be updated.

Type

array

3.1.34. .template.spec.containers[].env[]

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name of the environment variable. Must be a C_IDENTIFIER.

Property	Type	Description
value	string	Variable references <code>\$(VAR_NAME)</code> are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double <code>\$</code> are reduced to a single <code>\$</code> , which allows for escaping the <code>\$(VAR_NAME)</code> syntax: i.e. <code>"\$(VAR_NAME)"</code> will produce the string literal <code>"\$(VAR_NAME)"</code> . Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to <code>""</code> .
valueFrom	object	<code>EnvVarSource</code> represents a source for the value of an <code>EnvVar</code> .

3.1.35. `.template.spec.containers[].env[].valueFrom`

Description

`EnvVarSource` represents a source for the value of an `EnvVar`.

Type

object

Property	Type	Description
configMapKeyRef	object	Selects a key from a <code>ConfigMap</code> .
fieldRef	object	<code>ObjectFieldSelector</code> selects an <code>APIVersioned</code> field of an object.
resourceFieldRef	object	<code>ResourceFieldSelector</code> represents container resources (cpu, memory) and their output format
secretKeyRef	object	<code>SecretKeySelector</code> selects a key of a <code>Secret</code> .

3.1.36. `.template.spec.containers[].env[].valueFrom.configMapKeyRef`

Description

Selects a key from a `ConfigMap`.

Type

object

Required

- **key**

Property	Type	Description
key	string	The key to select.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the ConfigMap or its key must be defined

3.1.37. .template.spec.containers[].env[].valueFrom.fieldRef

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

- **fieldPath**

Property	Type	Description
apiVersion	string	Version of the schema the FieldPath is written in terms of, defaults to "v1".
fieldPath	string	Path of the field to select in the specified API version.

3.1.38. .template.spec.containers[].env[].valueFrom.resourceFieldRef

Description

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object**Required**

- **resource**

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	Quantity	Specifies the output format of the exposed resources, defaults to "1"
resource	string	Required: resource to select

3.1.39. .template.spec.containers[].env[].valueFrom.secretKeyRef**Description**

SecretKeySelector selects a key of a Secret.

Type**object****Required**

- **key**

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the Secret or its key must be defined

3.1.40. .template.spec.containers[].envFrom**Description**

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

3.1.41. .template.spec.containers[].envFrom[]

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

object

Property	Type	Description
configMapRef	object	<p>ConfigMapEnvSource selects a ConfigMap to populate the environment variables with.</p> <p>The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.</p>
prefix	string	<p>An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.</p>
secretRef	object	<p>SecretEnvSource selects a Secret to populate the environment variables with.</p> <p>The contents of the target Secret's Data field will represent the key-value pairs as environment variables.</p>

3.1.42. .template.spec.containers[].envFrom[].configMapRef

Description

ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the ConfigMap must be defined

3.1.43. .template.spec.containers[].envFrom[].secretRef

Description

SecretEnvSource selects a Secret to populate the environment variables with.

The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the Secret must be defined

3.1.44. .template.spec.containers[].lifecycle

Description

Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.

Type

object

Property	Type	Description
postStart	object	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCP Socket must be specified.
preStop	object	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCP Socket must be specified.

3.1.45. .template.spec.containers[].lifecycle.postStart

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCP Socket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCP SocketAction describes an action based on opening a socket

3.1.46. .template.spec.containers[].lifecycle.postStart.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
----------	------	-------------

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

3.1.47. .template.spec.containers[].lifecycle.postStart.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.

Property	Type	Description
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

3.1.48. .template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

3.1.49. .template.spec.containers[].lifecycle.postStart.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

3.1.50. .template.spec.containers[].lifecycle.postStart.sleep

Description

SleepAction describes a "sleep" action.

Type**object****Required**

- **seconds**

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

3.1.51. .template.spec.containers[].lifecycle.postStart.tcpSocket**Description**

TCPSocketAction describes an action based on opening a socket

Type**object****Required**

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

3.1.52. .template.spec.containers[].lifecycle.preStop**Description**

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

Type**object**

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.

Property	Type	Description
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

3.1.53. .template.spec.containers[].lifecycle.preStop.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

3.1.54. .template.spec.containers[].lifecycle.preStop.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

3.1.55. .template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

3.1.56. .template.spec.containers[].lifecycle.preStop.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

3.1.57. .template.spec.containers[].lifecycle.preStop.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

- **seconds**

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

3.1.58. .template.spec.containers[].lifecycle.preStop.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

3.1.59. .template.spec.containers[].livenessProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPC specifies an action involving a GRPC port.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

Property	Type	Description
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

3.1.60. .template.spec.containers[].livenessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
----------	------	-------------

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

3.1.61. .template.spec.containers[].livenessProbe.grpc

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

- **port**

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

3.1.62. .template.spec.containers[].livenessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

3.1.63. .template.spec.containers[].livenessProbe.httpGet.httpHeaders**Description**

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

3.1.64. .template.spec.containers[].livenessProbe.httpGet.httpHeaders[]**Description**

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

3.1.65. .template.spec.containers[].livenessProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

3.1.66. .template.spec.containers[].ports

Description

List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See <https://github.com/kubernetes/kubernetes/issues/108255>. Cannot be updated.

Type

array

3.1.67. .template.spec.containers[].ports[]

Description

ContainerPort represents a network port in a single container.

Type

object

Required

- **containerPort**

Property	Type	Description
containerPort	integer	Number of port to expose on the pod's IP address. This must be a valid port number, $0 < x < 65536$.
hostIP	string	What host IP to bind the external port to.
hostPort	integer	Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
name	string	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
protocol	string	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP". Possible enum values: - "SCTP" is the SCTP protocol. - "TCP" is the TCP protocol. - "UDP" is the UDP protocol.

3.1.68. .template.spec.containers[].readinessProbe**Description**

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPC specifies an action involving a GRPC port.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

Property	Type	Description
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

3.1.69. .template.spec.containers[].readinessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
----------	------	-------------

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

3.1.70. .template.spec.containers[].readinessProbe.grpc

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

- **port**

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

3.1.71. .template.spec.containers[].readinessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

3.1.72. .template.spec.containers[].readinessProbe.httpGet.httpHeaders**Description**

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

3.1.73. .template.spec.containers[].readinessProbe.httpGet.httpHeaders[]**Description**

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

3.1.74. .template.spec.containers[].readinessProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

3.1.75. .template.spec.containers[].resizePolicy

Description

Resources resize policy for the container.

Type

array

3.1.76. .template.spec.containers[].resizePolicy[]

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

object**Required**

- **resourceName**
- **restartPolicy**

Property	Type	Description
resourceName	string	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.
restartPolicy	string	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

3.1.77. .template.spec.containers[].resources**Description**

ResourceRequirements describes the compute resource requirements.

Type**object**

Property	Type	Description
claims	array	<p>Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container.</p> <p>This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.</p> <p>This field is immutable. It can only be set for containers.</p>
claims[]	object	ResourceClaim references one entry in PodSpec.ResourceClaims.
limits	object (Quantity)	<p>Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/</p>

Property	Type	Description
requests	object (Quantity)	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

3.1.78. .template.spec.containers[].resources.claims

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.

This field is immutable. It can only be set for containers.

Type

array

3.1.79. .template.spec.containers[].resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.

Property	Type	Description
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

3.1.80. .template.spec.containers[].securityContext

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

object

Property	Type	Description
allowPrivilegeEscalation	boolean	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the no_new_privs flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1) run as Privileged 2) has CAP_SYS_ADMIN Note that this field cannot be set when spec.os.name is windows.
appArmorProfile	object	AppArmorProfile defines a pod or container's AppArmor settings.
capabilities	object	Adds and removes POSIX capabilities from running containers.
privileged	boolean	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
procMount	string	<p>procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.</p> <p>Possible enum values: - "Default" uses the container runtime defaults for readonly and masked paths for /proc. Most container runtimes mask certain paths in /proc to avoid accidental security exposure of special devices or information. - "Unmasked" bypasses the default masking behavior of the container runtime and ensures the newly created /proc the container stays in tact with no modifications.</p>
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	SELinuxOptions are the labels to be applied to the container
seccompProfile	object	SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.
windowsOptions	object	WindowsSecurityContextOptions contain Windows-specific options and credentials.

3.1.81. .template.spec.containers[].securityContext.appArmorProfile

Description

AppArmorProfile defines a pod or container's AppArmor settings.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	<p>type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.</p> <p>Possible enum values: - "Localhost" indicates that a profile pre-loaded on the node should be used. - "RuntimeDefault" indicates that the container runtime's default AppArmor profile should be used. - "Unconfined" indicates that no AppArmor profile should be enforced.</p>

3.1.82. .template.spec.containers[].securityContext.capabilities

Description

Adds and removes POSIX capabilities from running containers.

Type

object

Property	Type	Description
add	array (string)	Added capabilities
drop	array (string)	Removed capabilities

3.1.83. .template.spec.containers[].securityContext.seLinuxOptions

Description

SELinuxOptions are the labels to be applied to the container

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

3.1.84. .template.spec.containers[].securityContext.seccompProfile

Description

SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.

Property	Type	Description
type	string	<p>type indicates which kind of seccomp profile will be applied. Valid options are:</p> <p>Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.</p> <p>Possible enum values: - "Localhost" indicates a profile defined in a file on the node should be used. The file's location relative to <kubelet-root-dir>/seccomp. - "RuntimeDefault" represents the default container runtime seccomp profile. - "Unconfined" indicates no seccomp profile is applied (A.K.A. unconfined).</p>

3.1.85. .template.spec.containers[].securityContext.windowsOptions

Description

WindowsSecurityContextOptions contain Windows-specific options and credentials.

Type

object

Property	Type	Description
gmsaCredentialSpec	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa) inlines the contents of the GMSA credential spec named by the GMSACredentialSpecName field.
gmsaCredentialSpecName	string	GMSACredentialSpecName is the name of the GMSA credential spec to use.

Property	Type	Description
hostProcess	boolean	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
runAsUserName	string	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

3.1.86. .template.spec.containers[].startupProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPC specifies an action involving a GRPC port.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.

Property	Type	Description
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's <code>terminationGracePeriodSeconds</code> will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling <code>ProbeTerminationGracePeriod</code> feature gate. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used if unset.

Property	Type	Description
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

3.1.87. .template.spec.containers[].startupProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

3.1.88. .template.spec.containers[].startupProbe.grpc

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

- **port**

Property	Type	Description
----------	------	-------------

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	<p>Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md).</p> <p>If this is not specified, the default behavior is defined by gRPC.</p>

3.1.89. .template.spec.containers[].startupProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

Property	Type	Description
scheme	string	<p>Scheme to use for connecting to the host. Defaults to HTTP.</p> <p>Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://</p>

3.1.90. .template.spec.containers[].startupProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

3.1.91. .template.spec.containers[].startupProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

3.1.92. .template.spec.containers[].startupProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

3.1.93. .template.spec.containers[].volumeDevices

Description

volumeDevices is the list of block devices to be used by the container.

Type

array

3.1.94. .template.spec.containers[].volumeDevices[]

Description

volumeDevice describes a mapping of a raw block device within a container.

Type

object

Required

- **name**
- **devicePath**

Property	Type	Description
devicePath	string	devicePath is the path inside of the container that the device will be mapped to.
name	string	name must match the name of a persistentVolumeClaim in the pod

3.1.95. .template.spec.containers[].volumeMounts

Description

Pod volumes to mount into the container's filesystem. Cannot be updated.

Type

array

3.1.96. .template.spec.containers[].volumeMounts[]

Description

VolumeMount describes a mounting of a Volume within a container.

Type

object

Required

- **name**
- **mountPath**

Property	Type	Description
mountPath	string	Path within the container at which the volume should be mounted. Must not contain ':

Property	Type	Description
mountPropagation	string	<p>mountPropagation determines how mounts are propagated from the host to container and the other way around. When not set, MountPropagationNone is used. This field is beta in 1.10. When RecursiveReadOnly is set to IfPossible or to Enabled, MountPropagation must be None or unspecified (which defaults to None).</p> <p>Possible enum values: - "Bidirectional" means that the volume in a container will receive new mounts from the host or other containers, and its own mounts will be propagated from the container to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rshared" in Linux terminology). - "HostToContainer" means that the volume in a container will receive new mounts from the host or other containers, but filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rslave" in Linux terminology). - "None" means that the volume in a container will not receive new mounts from the host or other containers, and filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode corresponds to "private" in Linux terminology.</p>
name	string	This must match the Name of a Volume.
readOnly	boolean	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.

Property	Type	Description
recursiveReadOnly	string	<p>RecursiveReadOnly specifies whether read-only mounts should be handled recursively.</p> <p>If ReadOnly is false, this field has no meaning and must be unspecified.</p> <p>If ReadOnly is true, and this field is set to Disabled, the mount is not made recursively read-only. If this field is set to IfPossible, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to Enabled, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason.</p> <p>If this field is set to IfPossible or Enabled, MountPropagation must be set to None (or be unspecified, which defaults to None).</p> <p>If this field is not specified, it is treated as an equivalent of Disabled.</p>
subPath	string	<p>Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).</p>
subPathExpr	string	<p>Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.</p>

3.1.97. .template.spec.dnsConfig

Description

PodDNSConfig defines the DNS parameters of a pod in addition to those generated from DNSPolicy.

Type**object**

Property	Type	Description
nameservers	array (string)	A list of DNS name server IP addresses. This will be appended to the base nameservers generated from DNSPolicy. Duplicated nameservers will be removed.
options	array	A list of DNS resolver options. This will be merged with the base options generated from DNSPolicy. Duplicated entries will be removed. Resolution options given in Options will override those that appear in the base DNSPolicy.
options[]	object	PodDNSConfigOption defines DNS resolver options of a pod.
searches	array (string)	A list of DNS search domains for host-name lookup. This will be appended to the base search paths generated from DNSPolicy. Duplicated search paths will be removed.

3.1.98. .template.spec.dnsConfig.options**Description**

A list of DNS resolver options. This will be merged with the base options generated from DNSPolicy. Duplicated entries will be removed. Resolution options given in Options will override those that appear in the base DNSPolicy.

Type**array****3.1.99. .template.spec.dnsConfig.options[]****Description**

PodDNSConfigOption defines DNS resolver options of a pod.

Type**object**

Property	Type	Description
name	string	Required.
value	string	

3.1.100. .template.spec.ephemeralContainers

Description

List of ephemeral containers run in this pod. Ephemeral containers may be run in an existing pod to perform user-initiated actions such as debugging. This list cannot be specified when creating a pod, and it cannot be modified by updating the pod spec. In order to add an ephemeral container to an existing pod, use the pod's ephemeralcontainers subresource.

Type

array

3.1.101. .template.spec.ephemeralContainers[]

Description

An EphemeralContainer is a temporary container that you may add to an existing Pod for user-initiated activities such as debugging. Ephemeral containers have no resource or scheduling guarantees, and they will not be restarted when they exit or when a Pod is removed or restarted. The kubelet may evict a Pod if an ephemeral container causes the Pod to exceed its resource allocation. To add an ephemeral container, use the ephemeralcontainers subresource of an existing Pod. Ephemeral containers may not be removed or restarted.

Type

object

Required

- **name**

Property	Type	Description
----------	------	-------------

Property	Type	Description
args	array (string)	<p>Arguments to the entrypoint. The image's CMD is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell</p>
command	array (string)	<p>Entrypoint array. Not executed within a shell. The image's ENTRYPOINT is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell</p>
env	array	<p>List of environment variables to set in the container. Cannot be updated.</p>

Property	Type	Description
env[]	object	EnvVar represents an environment variable present in a Container.
envFrom	array	List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
envFrom[]	object	EnvFromSource represents the source of a set of ConfigMaps
image	string	Container image name. More info: https://kubernetes.io/docs/concepts/containers/images
imagePullPolicy	string	<p>Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images#updating-images</p> <p>Possible enum values: - "Always" means that kubelet always attempts to pull the latest image. Container will fail If the pull fails. - "IfNotPresent" means that kubelet pulls if the image isn't present on disk. Container will fail if the image isn't present and the pull fails. - "Never" means that kubelet never pulls an image, but only uses a local image. Container will fail if the image isn't present</p>

Property	Type	Description
lifecycle	object	Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.
livenessProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
name	string	Name of the ephemeral container specified as a DNS_LABEL. This name must be unique among all containers, init containers and ephemeral containers.
ports	array	Ports are not allowed for ephemeral containers.
ports[]	object	ContainerPort represents a network port in a single container.
readinessProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
resizePolicy	array	Resources resize policy for the container.
resizePolicy[]	object	ContainerResizePolicy represents resource resize policy for the container.
resources	object	ResourceRequirements describes the compute resource requirements.

Property	Type	Description
restartPolicy	string	Restart policy for the container to manage the restart behavior of each container within a pod. This may only be set for init containers. You cannot set this field on ephemeral containers.
securityContext	object	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
startupProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
stdin	boolean	Whether this container should allocate a buffer for stdin in the container runtime. If this is not set, reads from stdin in the container will always result in EOF. Default is false.
stdinOnce	boolean	Whether the container runtime should close the stdin channel after it has been opened by a single attach. When stdin is true the stdin stream will remain open across multiple attach sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, and then remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container processes that reads from stdin will never receive an EOF. Default is false

Property	Type	Description
targetContainerName	string	<p>If set, the name of the container from PodSpec that this ephemeral container targets. The ephemeral container will be run in the namespaces (IPC, PID, etc) of this container. If not set then the ephemeral container uses the namespaces configured in the Pod spec.</p> <p>The container runtime must implement support for this feature. If the runtime does not support namespace targeting then the result of setting this field is undefined.</p>
terminationMessagePath	string	<p>Optional: Path at which the file to which the container's termination message will be written is mounted into the container's filesystem. Message written is intended to be brief final status, such as an assertion failure message. Will be truncated by the node if greater than 4096 bytes. The total message length across all containers will be limited to 12kb. Defaults to <code>/dev/termination-log</code>. Cannot be updated.</p>

Property	Type	Description
terminationMessagePolicy	string	<p>Indicate how the termination message should be populated. File will use the contents of terminationMessagePath to populate the container status message on both success and failure. FallbackToLogsOnError will use the last chunk of container log output if the termination message file is empty and the container exited with an error. The log output is limited to 2048 bytes or 80 lines, whichever is smaller. Defaults to File. Cannot be updated.</p> <p>Possible enum values: - "FallbackToLogsOnError" will read the most recent contents of the container logs for the container status message when the container exits with an error and the terminationMessagePath has no contents. - "File" is the default behavior and will set the container status message to the contents of the container's terminationMessagePath when the container exits.</p>
tty	boolean	Whether this container should allocate a TTY for itself, also requires 'stdin' to be true. Default is false.
volumeDevices	array	volumeDevices is the list of block devices to be used by the container.
volumeDevices[]	object	volumeDevice describes a mapping of a raw block device within a container.
volumeMounts	array	Pod volumes to mount into the container's filesystem. Subpath mounts are not allowed for ephemeral containers. Cannot be updated.

Property	Type	Description
volumeMounts[]	object	VolumeMount describes a mounting of a Volume within a container.
workingDir	string	Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image. Cannot be updated.

3.1.102. .template.spec.ephemeralContainers[].env

Description

List of environment variables to set in the container. Cannot be updated.

Type

array

3.1.103. .template.spec.ephemeralContainers[].env[]

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name of the environment variable. Must be a C_IDENTIFIER.

Property	Type	Description
value	string	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$ (VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
valueFrom	object	EnvVarSource represents a source for the value of an EnvVar.

3.1.104. .template.spec.ephemeralContainers[].env[].valueFrom

Description

EnvVarSource represents a source for the value of an EnvVar.

Type

object

Property	Type	Description
configMapKeyRef	object	Selects a key from a ConfigMap.
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.
resourceFieldRef	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format
secretKeyRef	object	SecretKeySelector selects a key of a Secret.

3.1.105. .template.spec.ephemeralContainers[].env[].valueFrom.configMapKeyRef

Description

Selects a key from a ConfigMap.

Type

object

Required

- **key**

Property	Type	Description
key	string	The key to select.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the ConfigMap or its key must be defined

3.1.106. .template.spec.ephemeralContainers[].env[].valueFrom.fieldRef

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

- **fieldPath**

Property	Type	Description
apiVersion	string	Version of the schema the FieldPath is written in terms of, defaults to "v1".
fieldPath	string	Path of the field to select in the specified API version.

3.1.107. .template.spec.ephemeralContainers[].env[].valueFrom.resourceFieldRef

Description

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

- **resource**

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	Quantity	Specifies the output format of the exposed resources, defaults to "1"
resource	string	Required: resource to select

3.1.108. .template.spec.ephemeralContainers[].env[].valueFrom.secretKeyRef

Description

SecretKeySelector selects a key of a Secret.

Type

object

Required

- **key**

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the Secret or its key must be defined

3.1.109. .template.spec.ephemeralContainers[].envFrom

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

3.1.110. .template.spec.ephemeralContainers[].envFrom[]**Description**

EnvFromSource represents the source of a set of ConfigMaps

Type

object

Property	Type	Description
configMapRef	object	<p>ConfigMapEnvSource selects a ConfigMap to populate the environment variables with.</p> <p>The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.</p>
prefix	string	<p>An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.</p>
secretRef	object	<p>SecretEnvSource selects a Secret to populate the environment variables with.</p> <p>The contents of the target Secret's Data field will represent the key-value pairs as environment variables.</p>

3.1.111. .template.spec.ephemeralContainers[].envFrom[].configMapRef**Description**

ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the ConfigMap must be defined

3.1.112. .template.spec.ephemeralContainers[].envFrom[].secretRef

Description

SecretEnvSource selects a Secret to populate the environment variables with.

The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the Secret must be defined

3.1.113. .template.spec.ephemeralContainers[].lifecycle

Description

Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.

Type

object

Property	Type	Description
postStart	object	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCP Socket must be specified.
preStop	object	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCP Socket must be specified.

3.1.114. .template.spec.ephemeralContainers[].lifecycle.postStart

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCP Socket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCP Socket Action describes an action based on opening a socket

3.1.115. .template.spec.ephemeralContainers[].lifecycle.postStart.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
----------	------	-------------

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

3.1.116. .template.spec.ephemeralContainers[].lifecycle.postStart.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.

Property	Type	Description
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

3.1.117. .template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeader:

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

3.1.118. .template.spec.ephemeralContainers[].lifecycle.postStart.httpGet.httpHeader:

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

3.1.119. .template.spec.ephemeralContainers[].lifecycle.postStart.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

- **seconds**

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

3.1.120. .template.spec.ephemeralContainers[].lifecycle.postStart.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

3.1.121. .template.spec.ephemeralContainers[].lifecycle.preStop

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.

Property	Type	Description
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

3.1.122. .template.spec.ephemeralContainers[].lifecycle.preStop.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

3.1.123. .template.spec.ephemeralContainers[].lifecycle.preStop.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

3.1.124. .template.spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

3.1.125. .template.spec.ephemeralContainers[].lifecycle.preStop.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

3.1.126. .template.spec.ephemeralContainers[].lifecycle.preStop.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

- **seconds**

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

3.1.127. .template.spec.ephemeralContainers[].lifecycle.preStop.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

3.1.128. .template.spec.ephemeralContainers[].livenessProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPC specifies an action involving a GRPC port.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

Property	Type	Description
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

3.1.129. .template.spec.ephemeralContainers[].livenessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
----------	------	-------------

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

3.1.130. .template.spec.ephemeralContainers[].livenessProbe.grpc

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

- **port**

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

3.1.131. .template.spec.ephemeralContainers[].livenessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

3.1.132. .template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders**Description**

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

3.1.133. .template.spec.ephemeralContainers[].livenessProbe.httpGet.httpHeaders[]**Description**

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

3.1.134. .template.spec.ephemeralContainers[].livenessProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

3.1.135. .template.spec.ephemeralContainers[].ports

Description

Ports are not allowed for ephemeral containers.

Type

array

3.1.136. .template.spec.ephemeralContainers[].ports[]

Description

ContainerPort represents a network port in a single container.

Type

object**Required**

- **containerPort**

Property	Type	Description
containerPort	integer	Number of port to expose on the pod's IP address. This must be a valid port number, $0 < x < 65536$.
hostIP	string	What host IP to bind the external port to.
hostPort	integer	Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
name	string	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
protocol	string	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP". Possible enum values: - "SCTP" is the SCTP protocol. - "TCP" is the TCP protocol. - "UDP" is the UDP protocol.

3.1.137. .template.spec.ephemeralContainers[].readinessProbe**Description**

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type**object**

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.

Property	Type	Description
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPC specifies an action involving a GRPC port.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

Property	Type	Description
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

3.1.138. .template.spec.ephemeralContainers[].readinessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
----------	------	-------------

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

3.1.139. .template.spec.ephemeralContainers[].readinessProbe.grpc

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

- **port**

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

3.1.140. .template.spec.ephemeralContainers[].readinessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

3.1.141. .template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders**Description**

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

3.1.142. .template.spec.ephemeralContainers[].readinessProbe.httpGet.httpHeaders[]**Description**

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

3.1.143. .template.spec.ephemeralContainers[].readinessProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

3.1.144. .template.spec.ephemeralContainers[].resizePolicy

Description

Resources resize policy for the container.

Type

array

3.1.145. .template.spec.ephemeralContainers[].resizePolicy[]

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

object**Required**

- **resourceName**
- **restartPolicy**

Property	Type	Description
resourceName	string	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.
restartPolicy	string	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

3.1.146. .template.spec.ephemeralContainers[].resources**Description**

ResourceRequirements describes the compute resource requirements.

Type**object**

Property	Type	Description
claims	array	<p>Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container.</p> <p>This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.</p> <p>This field is immutable. It can only be set for containers.</p>
claims[]	object	ResourceClaim references one entry in PodSpec.ResourceClaims.
limits	object (Quantity)	<p>Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/</p>

Property	Type	Description
requests	object (Quantity)	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

3.1.147. .template.spec.ephemeralContainers[].resources.claims

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.

This field is immutable. It can only be set for containers.

Type

array

3.1.148. .template.spec.ephemeralContainers[].resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.

Property	Type	Description
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

3.1.149. .template.spec.ephemeralContainers[].securityContext

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

object

Property	Type	Description
allowPrivilegeEscalation	boolean	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the no_new_privs flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1) run as Privileged 2) has CAP_SYS_ADMIN Note that this field cannot be set when spec.os.name is windows.
appArmorProfile	object	AppArmorProfile defines a pod or container's AppArmor settings.
capabilities	object	Adds and removes POSIX capabilities from running containers.
privileged	boolean	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
procMount	string	<p>procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.</p> <p>Possible enum values: - "Default" uses the container runtime defaults for readonly and masked paths for /proc. Most container runtimes mask certain paths in /proc to avoid accidental security exposure of special devices or information. - "Unmasked" bypasses the default masking behavior of the container runtime and ensures the newly created /proc the container stays in tact with no modifications.</p>
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	SELinuxOptions are the labels to be applied to the container
seccompProfile	object	SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.
windowsOptions	object	WindowsSecurityContextOptions contain Windows-specific options and credentials.

3.1.150. .template.spec.ephemeralContainers[].securityContext.appArmorProfile

Description

AppArmorProfile defines a pod or container's AppArmor settings.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	<p>type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.</p> <p>Possible enum values: - "Localhost" indicates that a profile pre-loaded on the node should be used. - "RuntimeDefault" indicates that the container runtime's default AppArmor profile should be used. - "Unconfined" indicates that no AppArmor profile should be enforced.</p>

3.1.151. .template.spec.ephemeralContainers[].securityContext.capabilities

Description

Adds and removes POSIX capabilities from running containers.

Type

object

Property	Type	Description
add	array (string)	Added capabilities
drop	array (string)	Removed capabilities

3.1.152. .template.spec.ephemeralContainers[].securityContext.seLinuxOptions

Description

SELinuxOptions are the labels to be applied to the container

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

3.1.153. .template.spec.ephemeralContainers[].securityContext.seccompProfile

Description

SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.

Property	Type	Description
type	string	<p>type indicates which kind of seccomp profile will be applied. Valid options are:</p> <p>Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.</p> <p>Possible enum values: - "Localhost" indicates a profile defined in a file on the node should be used. The file's location relative to <kubelet-root-dir>/seccomp. - "RuntimeDefault" represents the default container runtime seccomp profile. - "Unconfined" indicates no seccomp profile is applied (A.K.A. unconfined).</p>

3.1.154. .template.spec.ephemeralContainers[].securityContext.windowsOptions

Description

WindowsSecurityContextOptions contain Windows-specific options and credentials.

Type

object

Property	Type	Description
gmsaCredentialSpec	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa) inlines the contents of the GMSA credential spec named by the GMSACredentialSpecName field.
gmsaCredentialSpecName	string	GMSACredentialSpecName is the name of the GMSA credential spec to use.

Property	Type	Description
hostProcess	boolean	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
runAsUserName	string	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

3.1.155. .template.spec.ephemeralContainers[].startupProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPC specifies an action involving a GRPC port.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.

Property	Type	Description
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's <code>terminationGracePeriodSeconds</code> will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling <code>ProbeTerminationGracePeriod</code> feature gate. Minimum value is 1. <code>spec.terminationGracePeriodSeconds</code> is used if unset.

Property	Type	Description
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

3.1.156. .template.spec.ephemeralContainers[].startupProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

3.1.157. .template.spec.ephemeralContainers[].startupProbe.grpc

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

- **port**

Property	Type	Description
----------	------	-------------

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

3.1.158. .template.spec.ephemeralContainers[].startupProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

Property	Type	Description
scheme	string	<p>Scheme to use for connecting to the host. Defaults to HTTP.</p> <p>Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://</p>

3.1.159. .template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

3.1.160. .template.spec.ephemeralContainers[].startupProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

3.1.161. .template.spec.ephemeralContainers[].startupProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

3.1.162. .template.spec.ephemeralContainers[].volumeDevices

Description

volumeDevices is the list of block devices to be used by the container.

Type

array

3.1.163. .template.spec.ephemeralContainers[].volumeDevices[]

Description

volumeDevice describes a mapping of a raw block device within a container.

Type

object

Required

- **name**
- **devicePath**

Property	Type	Description
devicePath	string	devicePath is the path inside of the container that the device will be mapped to.
name	string	name must match the name of a persistentVolumeClaim in the pod

3.1.164. .template.spec.ephemeralContainers[].volumeMounts

Description

Pod volumes to mount into the container's filesystem. Subpath mounts are not allowed for ephemeral containers. Cannot be updated.

Type

array

3.1.165. `.template.spec.ephemeralContainers[].volumeMounts[]`

Description

VolumeMount describes a mounting of a Volume within a container.

Type

object

Required

- **name**
- **mountPath**

Property	Type	Description
mountPath	string	Path within the container at which the volume should be mounted. Must not contain ':

Property	Type	Description
mountPropagation	string	<p>mountPropagation determines how mounts are propagated from the host to container and the other way around. When not set, MountPropagationNone is used. This field is beta in 1.10. When RecursiveReadOnly is set to IfPossible or to Enabled, MountPropagation must be None or unspecified (which defaults to None).</p> <p>Possible enum values: - "Bidirectional" means that the volume in a container will receive new mounts from the host or other containers, and its own mounts will be propagated from the container to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rshared" in Linux terminology). - "HostToContainer" means that the volume in a container will receive new mounts from the host or other containers, but filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rslave" in Linux terminology). - "None" means that the volume in a container will not receive new mounts from the host or other containers, and filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode corresponds to "private" in Linux terminology.</p>
name	string	This must match the Name of a Volume.
readOnly	boolean	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.

Property	Type	Description
recursiveReadOnly	string	<p>RecursiveReadOnly specifies whether read-only mounts should be handled recursively.</p> <p>If ReadOnly is false, this field has no meaning and must be unspecified.</p> <p>If ReadOnly is true, and this field is set to Disabled, the mount is not made recursively read-only. If this field is set to IfPossible, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to Enabled, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason.</p> <p>If this field is set to IfPossible or Enabled, MountPropagation must be set to None (or be unspecified, which defaults to None).</p> <p>If this field is not specified, it is treated as an equivalent of Disabled.</p>
subPath	string	<p>Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).</p>
subPathExpr	string	<p>Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.</p>

3.1.166. .template.spec.hostAliases

Description

HostAliases is an optional list of hosts and IPs that will be injected into the pod's hosts file if specified.

Type

array

3.1.167. .template.spec.hostAliases[]

Description

HostAlias holds the mapping between IP and hostnames that will be injected as an entry in the pod's hosts file.

Type

object

Required

- **ip**

Property	Type	Description
hostnames	array (string)	Hostnames for the above IP address.
ip	string	IP address of the host file entry.

3.1.168. .template.spec.imagePullSecrets

Description

ImagePullSecrets is an optional list of references to secrets in the same namespace to use for pulling any of the images used by this PodSpec. If specified, these secrets will be passed to individual puller implementations for them to use. More info:

<https://kubernetes.io/docs/concepts/containers/images#specifying-imagepullsecrets-on-a-pod>

Type

array

3.1.169. .template.spec.imagePullSecrets[]

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
----------	------	-------------

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

3.1.170. .template.spec.initContainers

Description

List of initialization containers belonging to the pod. Init containers are executed in order prior to containers being started. If any init container fails, the pod is considered to have failed and is handled according to its restartPolicy. The name for an init container or normal container must be unique among all containers. Init containers may not have Lifecycle actions, Readiness probes, Liveness probes, or Startup probes. The resourceRequirements of an init container are taken into account during scheduling by finding the highest request/limit for each resource type, and then using the max of that value or the sum of the normal containers. Limits are applied to init containers in a similar fashion. Init containers cannot currently be added or removed. Cannot be updated. More info: <https://kubernetes.io/docs/concepts/workloads/pods/init-containers/>

Type

array

3.1.171. .template.spec.initContainers[]

Description

A single application container that you want to run within a pod.

Type

object

Required

- **name**

Property	Type	Description
----------	------	-------------

Property	Type	Description
args	array (string)	<p>Arguments to the endpoint. The container image's CMD is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell</p>
command	array (string)	<p>Entrypoint array. Not executed within a shell. The container image's ENTRYPOINT is used if this is not provided. Variable references \$(VAR_NAME) are expanded using the container's environment. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$(VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Cannot be updated. More info: https://kubernetes.io/docs/tasks/inject-data-application/define-command-argument-container/#running-a-command-in-a-shell</p>
env	array	<p>List of environment variables to set in the container. Cannot be updated.</p>

Property	Type	Description
env[]	object	EnvVar represents an environment variable present in a Container.
envFrom	array	List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.
envFrom[]	object	EnvFromSource represents the source of a set of ConfigMaps
image	string	Container image name. More info: https://kubernetes.io/docs/concepts/containers/images This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.

Property	Type	Description
imagePullPolicy	string	<p>Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated. More info: https://kubernetes.io/docs/concepts/containers/images#updating-images</p> <p>Possible enum values: - "Always" means that kubelet always attempts to pull the latest image. Container will fail if the pull fails. - "IfNotPresent" means that kubelet pulls if the image isn't present on disk. Container will fail if the image isn't present and the pull fails. - "Never" means that kubelet never pulls an image, but only uses a local image. Container will fail if the image isn't present</p>
lifecycle	object	<p>Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.</p>
livenessProbe	object	<p>Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.</p>
name	string	<p>Name of the container specified as a DNS_LABEL. Each container in a pod must have a unique name (DNS_LABEL). Cannot be updated.</p>

Property	Type	Description
ports	array	List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See https://github.com/kubernetes/kubernetes/issues/108255 . Cannot be updated.
ports[]	object	ContainerPort represents a network port in a single container.
readinessProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.
resizePolicy	array	Resources resize policy for the container.
resizePolicy[]	object	ContainerResizePolicy represents resource resize policy for the container.
resources	object	ResourceRequirements describes the compute resource requirements.

Property	Type	Description
restartPolicy	string	RestartPolicy defines the restart behavior of individual containers in a pod. This field may only be set for init containers, and the only allowed value is "Always". For non-init containers or when this field is not specified, the restart behavior is defined by the Pod's restart policy and the container type. Setting the RestartPolicy as "Always" for the init container will have the following effect: this init container will be continually restarted on exit until all regular containers have terminated. Once all regular containers have completed, all init containers with restartPolicy "Always" will be shut down. This lifecycle differs from normal init containers and is often referred to as a "sidecar" container. Although this init container still starts in the init container sequence, it does not wait for the container to complete before proceeding to the next init container. Instead, the next init container starts immediately after this init container is started, or after any startupProbe has successfully completed.
securityContext	object	SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.
startupProbe	object	Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Property	Type	Description
stdin	boolean	Whether this container should allocate a buffer for stdin in the container runtime. If this is not set, reads from stdin in the container will always result in EOF. Default is false.
stdinOnce	boolean	Whether the container runtime should close the stdin channel after it has been opened by a single attach. When stdin is true the stdin stream will remain open across multiple attach sessions. If stdinOnce is set to true, stdin is opened on container start, is empty until the first client attaches to stdin, and then remains open and accepts data until the client disconnects, at which time stdin is closed and remains closed until the container is restarted. If this flag is false, a container processes that reads from stdin will never receive an EOF. Default is false
terminationMessagePath	string	Optional: Path at which the file to which the container's termination message will be written is mounted into the container's filesystem. Message written is intended to be brief final status, such as an assertion failure message. Will be truncated by the node if greater than 4096 bytes. The total message length across all containers will be limited to 12kb. Defaults to /dev/termination-log. Cannot be updated.

Property	Type	Description
terminationMessagePolicy	string	<p>Indicate how the termination message should be populated. File will use the contents of terminationMessagePath to populate the container status message on both success and failure. FallbackToLogsOnError will use the last chunk of container log output if the termination message file is empty and the container exited with an error. The log output is limited to 2048 bytes or 80 lines, whichever is smaller. Defaults to File. Cannot be updated.</p> <p>Possible enum values: - "FallbackToLogsOnError" will read the most recent contents of the container logs for the container status message when the container exits with an error and the terminationMessagePath has no contents. - "File" is the default behavior and will set the container status message to the contents of the container's terminationMessagePath when the container exits.</p>
tty	boolean	Whether this container should allocate a TTY for itself, also requires 'stdin' to be true. Default is false.
volumeDevices	array	volumeDevices is the list of block devices to be used by the container.
volumeDevices[]	object	volumeDevice describes a mapping of a raw block device within a container.
volumeMounts	array	Pod volumes to mount into the container's filesystem. Cannot be updated.
volumeMounts[]	object	VolumeMount describes a mounting of a Volume within a container.

Property	Type	Description
workingDir	string	Container's working directory. If not specified, the container runtime's default will be used, which might be configured in the container image. Cannot be updated.

3.1.172. .template.spec.initContainers[].env

Description

List of environment variables to set in the container. Cannot be updated.

Type

array

3.1.173. .template.spec.initContainers[].env[]

Description

EnvVar represents an environment variable present in a Container.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name of the environment variable. Must be a C_IDENTIFIER.

Property	Type	Description
value	string	Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables. If a variable cannot be resolved, the reference in the input string will be unchanged. Double are reduced to a single \$, which allows for escaping the \$(VAR_NAME) syntax: i.e. "\$(VAR_NAME)" will produce the string literal "\$ (VAR_NAME)". Escaped references will never be expanded, regardless of whether the variable exists or not. Defaults to "".
valueFrom	object	EnvVarSource represents a source for the value of an EnvVar.

3.1.174. .template.spec.initContainers[].env[].valueFrom

Description

EnvVarSource represents a source for the value of an EnvVar.

Type

object

Property	Type	Description
configMapKeyRef	object	Selects a key from a ConfigMap.
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.
resourceFieldRef	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format
secretKeyRef	object	SecretKeySelector selects a key of a Secret.

3.1.175. .template.spec.initContainers[].env[].valueFrom.configMapKeyRef

Description

Selects a key from a ConfigMap.

Type**object****Required**

- **key**

Property	Type	Description
key	string	The key to select.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the ConfigMap or its key must be defined

3.1.176. .template.spec.initContainers[].env[].valueFrom.fieldRef**Description**

ObjectFieldSelector selects an APIVersioned field of an object.

Type**object****Required**

- **fieldPath**

Property	Type	Description
apiVersion	string	Version of the schema the FieldPath is written in terms of, defaults to "v1".
fieldPath	string	Path of the field to select in the specified API version.

3.1.177. .template.spec.initContainers[].env[].valueFrom.resourceFieldRef**Description**

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

- **resource**

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	Quantity	Specifies the output format of the exposed resources, defaults to "1"
resource	string	Required: resource to select

3.1.178. .template.spec.initContainers[].env[].valueFrom.secretKeyRef

Description

SecretKeySelector selects a key of a Secret.

Type

object

Required

- **key**

Property	Type	Description
key	string	The key of the secret to select from. Must be a valid secret key.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the Secret or its key must be defined

3.1.179. .template.spec.initContainers[].envFrom

Description

List of sources to populate environment variables in the container. The keys defined within a source must be a C_IDENTIFIER. All invalid keys will be reported as an event when the container is starting. When a key exists in multiple sources, the value associated with the last source will take precedence. Values defined by an Env with a duplicate key will take precedence. Cannot be updated.

Type

array

3.1.180. .template.spec.initContainers[].envFrom[]

Description

EnvFromSource represents the source of a set of ConfigMaps

Type

object

Property	Type	Description
configMapRef	object	<p>ConfigMapEnvSource selects a ConfigMap to populate the environment variables with.</p> <p>The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.</p>
prefix	string	<p>An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.</p>
secretRef	object	<p>SecretEnvSource selects a Secret to populate the environment variables with.</p> <p>The contents of the target Secret's Data field will represent the key-value pairs as environment variables.</p>

3.1.181. .template.spec.initContainers[].envFrom[].configMapRef

Description

ConfigMapEnvSource selects a ConfigMap to populate the environment variables with. The contents of the target ConfigMap's Data field will represent the key-value pairs as environment variables.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the ConfigMap must be defined

3.1.182. .template.spec.initContainers[].envFrom[].secretRef

Description

SecretEnvSource selects a Secret to populate the environment variables with.

The contents of the target Secret's Data field will represent the key-value pairs as environment variables.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	Specify whether the Secret must be defined

3.1.183. .template.spec.initContainers[].lifecycle

Description

Lifecycle describes actions that the management system should take in response to container lifecycle events. For the PostStart and PreStop lifecycle handlers, management of the container blocks until the action is complete, unless the container process fails, in which case the handler is aborted.

Type

object

Property	Type	Description
postStart	object	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCP Socket must be specified.
preStop	object	LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCP Socket must be specified.

3.1.184. .template.spec.initContainers[].lifecycle.postStart

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCP Socket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCP SocketAction describes an action based on opening a socket

3.1.185. .template.spec.initContainers[].lifecycle.postStart.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
----------	------	-------------

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

3.1.186. .template.spec.initContainers[].lifecycle.postStart.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.

Property	Type	Description
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

3.1.187. .template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

3.1.188. .template.spec.initContainers[].lifecycle.postStart.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

3.1.189. .template.spec.initContainers[].lifecycle.postStart.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

- **seconds**

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

3.1.190. .template.spec.initContainers[].lifecycle.postStart.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

3.1.191. .template.spec.initContainers[].lifecycle.preStop

Description

LifecycleHandler defines a specific action that should be taken in a lifecycle hook. One and only one of the fields, except TCPSocket must be specified.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.

Property	Type	Description
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
sleep	object	SleepAction describes a "sleep" action.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

3.1.192. .template.spec.initContainers[].lifecycle.preStop.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

3.1.193. .template.spec.initContainers[].lifecycle.preStop.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	Scheme to use for connecting to the host. Defaults to HTTP. Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://

3.1.194. .template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

3.1.195. .template.spec.initContainers[].lifecycle.preStop.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

3.1.196. .template.spec.initContainers[].lifecycle.preStop.sleep

Description

SleepAction describes a "sleep" action.

Type

object

Required

- **seconds**

Property	Type	Description
seconds	integer	Seconds is the number of seconds to sleep.

3.1.197. .template.spec.initContainers[].lifecycle.preStop.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

3.1.198. .template.spec.initContainers[].livenessProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPC specifies an action involving a GRPC port.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

Property	Type	Description
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

3.1.199. .template.spec.initContainers[].livenessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
----------	------	-------------

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

3.1.200. .template.spec.initContainers[].livenessProbe.grpc

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

- **port**

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

3.1.201. .template.spec.initContainers[].livenessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	<p>Scheme to use for connecting to the host. Defaults to HTTP.</p> <p>Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://</p>

3.1.202. .template.spec.initContainers[].livenessProbe.httpGet.httpHeaders**Description**

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

3.1.203. .template.spec.initContainers[].livenessProbe.httpGet.httpHeaders[]**Description**

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

3.1.204. .template.spec.initContainers[].livenessProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

3.1.205. .template.spec.initContainers[].ports

Description

List of ports to expose from the container. Not specifying a port here DOES NOT prevent that port from being exposed. Any port which is listening on the default "0.0.0.0" address inside a container will be accessible from the network. Modifying this array with strategic merge patch may corrupt the data. For more information See <https://github.com/kubernetes/kubernetes/issues/108255>. Cannot be updated.

Type

array

3.1.206. .template.spec.initContainers[].ports[]

Description

ContainerPort represents a network port in a single container.

Type

object

Required

- **containerPort**

Property	Type	Description
containerPort	integer	Number of port to expose on the pod's IP address. This must be a valid port number, $0 < x < 65536$.
hostIP	string	What host IP to bind the external port to.
hostPort	integer	Number of port to expose on the host. If specified, this must be a valid port number, $0 < x < 65536$. If HostNetwork is specified, this must match ContainerPort. Most containers do not need this.
name	string	If specified, this must be an IANA_SVC_NAME and unique within the pod. Each named port in a pod must have a unique name. Name for the port that can be referred to by services.
protocol	string	Protocol for port. Must be UDP, TCP, or SCTP. Defaults to "TCP". Possible enum values: - "SCTP" is the SCTP protocol. - "TCP" is the TCP protocol. - "UDP" is the UDP protocol.

3.1.207. .template.spec.initContainers[].readinessProbe**Description**

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPC specifies an action involving a GRPC port.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket

Property	Type	Description
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

3.1.208. .template.spec.initContainers[].readinessProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
----------	------	-------------

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

3.1.209. .template.spec.initContainers[].readinessProbe.grpc

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

- **port**

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

3.1.210. .template.spec.initContainers[].readinessProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.
scheme	string	<p>Scheme to use for connecting to the host. Defaults to HTTP.</p> <p>Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://</p>

3.1.211. .template.spec.initContainers[].readinessProbe.httpGet.httpHeaders**Description**

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

3.1.212. .template.spec.initContainers[].readinessProbe.httpGet.httpHeaders[]**Description**

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

3.1.213. .template.spec.initContainers[].readinessProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

3.1.214. .template.spec.initContainers[].resizePolicy

Description

Resources resize policy for the container.

Type

array

3.1.215. .template.spec.initContainers[].resizePolicy[]

Description

ContainerResizePolicy represents resource resize policy for the container.

Type

object

Required

- **resourceName**
- **restartPolicy**

Property	Type	Description
resourceName	string	Name of the resource to which this resource resize policy applies. Supported values: cpu, memory.
restartPolicy	string	Restart policy to apply when specified resource is resized. If not specified, it defaults to NotRequired.

3.1.216. .template.spec.initContainers[].resources

Description

ResourceRequirements describes the compute resource requirements.

Type

object

Property	Type	Description
claims	array	<p>Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container.</p> <p>This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.</p> <p>This field is immutable. It can only be set for containers.</p>
claims[]	object	ResourceClaim references one entry in PodSpec.ResourceClaims.
limits	object (Quantity)	<p>Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/</p>

Property	Type	Description
requests	object (Quantity)	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

3.1.217. .template.spec.initContainers[].resources.claims

Description

Claims lists the names of resources, defined in spec.resourceClaims, that are used by this container. This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.

This field is immutable. It can only be set for containers.

Type

array

3.1.218. .template.spec.initContainers[].resources.claims[]

Description

ResourceClaim references one entry in PodSpec.ResourceClaims.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name must match the name of one entry in pod.spec.resourceClaims of the Pod where this field is used. It makes that resource available inside a container.

Property	Type	Description
request	string	Request is the name chosen for a request in the referenced claim. If empty, everything from the claim is made available, otherwise only the result of this request.

3.1.219. .template.spec.initContainers[].securityContext

Description

SecurityContext holds security configuration that will be applied to a container. Some fields are present in both SecurityContext and PodSecurityContext. When both are set, the values in SecurityContext take precedence.

Type

object

Property	Type	Description
allowPrivilegeEscalation	boolean	AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process. This bool directly controls if the no_new_privs flag will be set on the container process. AllowPrivilegeEscalation is true always when the container is: 1) run as Privileged 2) has CAP_SYS_ADMIN Note that this field cannot be set when spec.os.name is windows.
appArmorProfile	object	AppArmorProfile defines a pod or container's AppArmor settings.
capabilities	object	Adds and removes POSIX capabilities from running containers.
privileged	boolean	Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
procMount	string	<p>procMount denotes the type of proc mount to use for the containers. The default value is Default which uses the container runtime defaults for readonly paths and masked paths. This requires the ProcMountType feature flag to be enabled. Note that this field cannot be set when spec.os.name is windows.</p> <p>Possible enum values: - "Default" uses the container runtime defaults for readonly and masked paths for /proc. Most container runtimes mask certain paths in /proc to avoid accidental security exposure of special devices or information. - "Unmasked" bypasses the default masking behavior of the container runtime and ensures the newly created /proc the container stays in tact with no modifications.</p>
readOnlyRootFilesystem	boolean	Whether this container has a read-only root filesystem. Default is false. Note that this field cannot be set when spec.os.name is windows.
runAsGroup	integer	The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.

Property	Type	Description
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	SELinuxOptions are the labels to be applied to the container
seccompProfile	object	SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.
windowsOptions	object	WindowsSecurityContextOptions contain Windows-specific options and credentials.

3.1.220. .template.spec.initContainers[].securityContext.appArmorProfile

Description

AppArmorProfile defines a pod or container's AppArmor settings.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".
type	string	<p>type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.</p> <p>Possible enum values: - "Localhost" indicates that a profile pre-loaded on the node should be used. - "RuntimeDefault" indicates that the container runtime's default AppArmor profile should be used. - "Unconfined" indicates that no AppArmor profile should be enforced.</p>

3.1.221. .template.spec.initContainers[].securityContext.capabilities

Description

Adds and removes POSIX capabilities from running containers.

Type

object

Property	Type	Description
add	array (string)	Added capabilities
drop	array (string)	Removed capabilities

3.1.222. .template.spec.initContainers[].securityContext.seLinuxOptions

Description

SELinuxOptions are the labels to be applied to the container

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

3.1.223. .template.spec.initContainers[].securityContext.seccompProfile

Description

SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.

Property	Type	Description
type	string	<p>type indicates which kind of seccomp profile will be applied. Valid options are:</p> <p>Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.</p> <p>Possible enum values: - "Localhost" indicates a profile defined in a file on the node should be used. The file's location relative to <kubelet-root-dir>/seccomp. - "RuntimeDefault" represents the default container runtime seccomp profile. - "Unconfined" indicates no seccomp profile is applied (A.K.A. unconfined).</p>

3.1.224. .template.spec.initContainers[].securityContext.windowsOptions

Description

WindowsSecurityContextOptions contain Windows-specific options and credentials.

Type

object

Property	Type	Description
gmsaCredentialSpec	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa) inlines the contents of the GMSA credential spec named by the GMSACredentialSpecName field.
gmsaCredentialSpecName	string	GMSACredentialSpecName is the name of the GMSA credential spec to use.

Property	Type	Description
hostProcess	boolean	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
runAsUserName	string	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

3.1.225. .template.spec.initContainers[].startupProbe

Description

Probe describes a health check to be performed against a container to determine whether it is alive or ready to receive traffic.

Type

object

Property	Type	Description
exec	object	ExecAction describes a "run in container" action.
failureThreshold	integer	Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
grpc	object	GRPC specifies an action involving a GRPC port.
httpGet	object	HTTPGetAction describes an action based on HTTP Get requests.

Property	Type	Description
initialDelaySeconds	integer	Number of seconds after the container has started before liveness probes are initiated. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes
periodSeconds	integer	How often (in seconds) to perform the probe. Default to 10 seconds. Minimum value is 1.
successThreshold	integer	Minimum consecutive successes for the probe to be considered successful after having failed. Defaults to 1. Must be 1 for liveness and startup. Minimum value is 1.
tcpSocket	object	TCPSocketAction describes an action based on opening a socket
terminationGracePeriodSeconds	integer	Optional duration in seconds the pod needs to terminate gracefully upon probe failure. The grace period is the duration in seconds after the processes running in the pod are sent a termination signal and the time when the processes are forcibly halted with a kill signal. Set this value longer than the expected cleanup time for your process. If this value is nil, the pod's terminationGracePeriodSeconds will be used. Otherwise, this value overrides the value provided by the pod spec. Value must be non-negative integer. The value zero indicates stop immediately via the kill signal (no opportunity to shut down). This is a beta field and requires enabling ProbeTerminationGracePeriod feature gate. Minimum value is 1. spec.terminationGracePeriodSeconds is used if unset.

Property	Type	Description
timeoutSeconds	integer	Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1. More info: https://kubernetes.io/docs/concepts/workloads/pods/pod-lifecycle#container-probes

3.1.226. .template.spec.initContainers[].startupProbe.exec

Description

ExecAction describes a "run in container" action.

Type

object

Property	Type	Description
command	array (string)	Command is the command line to execute inside the container, the working directory for the command is root ('/') in the container's filesystem. The command is simply exec'd, it is not run inside a shell, so traditional shell instructions (' ', etc) won't work. To use a shell, you need to explicitly call out to that shell. Exit status of 0 is treated as live/healthy and non-zero is unhealthy.

3.1.227. .template.spec.initContainers[].startupProbe.grpc

Description

GRPC specifies an action involving a GRPC port.

Type

object

Required

- **port**

Property	Type	Description
----------	------	-------------

Property	Type	Description
port	integer	Port number of the gRPC service. Number must be in the range 1 to 65535.
service	string	Service is the name of the service to place in the gRPC HealthCheckRequest (see https://github.com/grpc/grpc/blob/master/doc/health-checking.md). If this is not specified, the default behavior is defined by gRPC.

3.1.228. .template.spec.initContainers[].startupProbe.httpGet

Description

HTTPGetAction describes an action based on HTTP Get requests.

Type

object

Required

- **port**

Property	Type	Description
host	string	Host name to connect to, defaults to the pod IP. You probably want to set "Host" in httpHeaders instead.
httpHeaders	array	Custom headers to set in the request. HTTP allows repeated headers.
httpHeaders[]	object	HTTPHeader describes a custom header to be used in HTTP probes
path	string	Path to access on the HTTP server.
port	IntOrString	Name or number of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

Property	Type	Description
scheme	string	<p>Scheme to use for connecting to the host. Defaults to HTTP.</p> <p>Possible enum values: - "HTTP" means that the scheme used will be http:// - "HTTPS" means that the scheme used will be https://</p>

3.1.229. .template.spec.initContainers[].startupProbe.httpGet.httpHeaders

Description

Custom headers to set in the request. HTTP allows repeated headers.

Type

array

3.1.230. .template.spec.initContainers[].startupProbe.httpGet.httpHeaders[]

Description

HTTPHeader describes a custom header to be used in HTTP probes

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	The header field name. This will be canonicalized upon output, so case-variant names will be understood as the same header.
value	string	The header field value

3.1.231. .template.spec.initContainers[].startupProbe.tcpSocket

Description

TCPSocketAction describes an action based on opening a socket

Type

object

Required

- **port**

Property	Type	Description
host	string	Optional: Host name to connect to, defaults to the pod IP.
port	IntOrString	Number or name of the port to access on the container. Number must be in the range 1 to 65535. Name must be an IANA_SVC_NAME.

3.1.232. .template.spec.initContainers[].volumeDevices

Description

volumeDevices is the list of block devices to be used by the container.

Type

array

3.1.233. .template.spec.initContainers[].volumeDevices[]

Description

volumeDevice describes a mapping of a raw block device within a container.

Type

object

Required

- **name**
- **devicePath**

Property	Type	Description
devicePath	string	devicePath is the path inside of the container that the device will be mapped to.
name	string	name must match the name of a persistentVolumeClaim in the pod

3.1.234. .template.spec.initContainers[].volumeMounts

Description

Pod volumes to mount into the container's filesystem. Cannot be updated.

Type

array

3.1.235. .template.spec.initContainers[].volumeMounts[]

Description

VolumeMount describes a mounting of a Volume within a container.

Type

object

Required

- **name**
- **mountPath**

Property	Type	Description
mountPath	string	Path within the container at which the volume should be mounted. Must not contain ':

Property	Type	Description
mountPropagation	string	<p>mountPropagation determines how mounts are propagated from the host to container and the other way around. When not set, MountPropagationNone is used. This field is beta in 1.10. When RecursiveReadOnly is set to IfPossible or to Enabled, MountPropagation must be None or unspecified (which defaults to None).</p> <p>Possible enum values: - "Bidirectional" means that the volume in a container will receive new mounts from the host or other containers, and its own mounts will be propagated from the container to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rshared" in Linux terminology). - "HostToContainer" means that the volume in a container will receive new mounts from the host or other containers, but filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode is recursively applied to all mounts in the volume ("rslave" in Linux terminology). - "None" means that the volume in a container will not receive new mounts from the host or other containers, and filesystems mounted inside the container won't be propagated to the host or other containers. Note that this mode corresponds to "private" in Linux terminology.</p>
name	string	This must match the Name of a Volume.
readOnly	boolean	Mounted read-only if true, read-write otherwise (false or unspecified). Defaults to false.

Property	Type	Description
recursiveReadOnly	string	<p>RecursiveReadOnly specifies whether read-only mounts should be handled recursively.</p> <p>If ReadOnly is false, this field has no meaning and must be unspecified.</p> <p>If ReadOnly is true, and this field is set to Disabled, the mount is not made recursively read-only. If this field is set to IfPossible, the mount is made recursively read-only, if it is supported by the container runtime. If this field is set to Enabled, the mount is made recursively read-only if it is supported by the container runtime, otherwise the pod will not be started and an error will be generated to indicate the reason.</p> <p>If this field is set to IfPossible or Enabled, MountPropagation must be set to None (or be unspecified, which defaults to None).</p> <p>If this field is not specified, it is treated as an equivalent of Disabled.</p>
subPath	string	<p>Path within the volume from which the container's volume should be mounted. Defaults to "" (volume's root).</p>
subPathExpr	string	<p>Expanded path within the volume from which the container's volume should be mounted. Behaves similarly to SubPath but environment variable references \$(VAR_NAME) are expanded using the container's environment. Defaults to "" (volume's root). SubPathExpr and SubPath are mutually exclusive.</p>

3.1.236. .template.spec.os

Description

PodOS defines the OS parameters of a pod.

Type**object****Required**

- **name**

Property	Type	Description
name	string	Name is the name of the operating system. The currently supported values are linux and windows. Additional value may be defined in future and can be one of: https://github.com/opencontainers/runtime-spec/blob/master/config.md#platform-specific-configuration Clients should expect to handle additional values and treat unrecognized values in this field as os: null

3.1.237. .template.spec.readinessGates

Description

If specified, all readiness gates will be evaluated for pod readiness. A pod is ready when all its containers are ready AND all conditions specified in the readiness gates have status equal to "True"
More info: <https://git.k8s.io/enhancements/keps/sig-network/580-pod-readiness-gates>

Type**array**

3.1.238. .template.spec.readinessGates[]

Description

PodReadinessGate contains the reference to a pod condition

Type**object****Required**

- **conditionType**

Property	Type	Description
conditionType	string	ConditionType refers to a condition in the pod's condition list with matching type.

3.1.239. .template.spec.resourceClaims

Description

ResourceClaims defines which ResourceClaims must be allocated and reserved before the Pod is allowed to start. The resources will be made available to those containers which consume them by name.

This is an alpha field and requires enabling the DynamicResourceAllocation feature gate.

This field is immutable.

Type

array

3.1.240. .template.spec.resourceClaims[]

Description

PodResourceClaim references exactly one ResourceClaim, either directly or by naming a ResourceClaimTemplate which is then turned into a ResourceClaim for the pod.

It adds a name to it that uniquely identifies the ResourceClaim inside the Pod. Containers that need access to the ResourceClaim reference it with this name.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name uniquely identifies this resource claim inside the pod. This must be a DNS_LABEL.
resourceClaimName	string	ResourceClaimName is the name of a ResourceClaim object in the same namespace as this pod. Exactly one of ResourceClaimName and ResourceClaimTemplateName must be set.

Property	Type	Description
resourceClaimTemplateName	string	<p>ResourceClaimTemplateName is the name of a ResourceClaimTemplate object in the same namespace as this pod.</p> <p>The template will be used to create a new ResourceClaim, which will be bound to this pod. When this pod is deleted, the ResourceClaim will also be deleted. The pod name and resource name, along with a generated component, will be used to form a unique name for the ResourceClaim, which will be recorded in <code>pod.status.resourceClaimStatus</code>.</p> <p>This field is immutable and no changes will be made to the corresponding ResourceClaim by the control plane after creating the ResourceClaim.</p> <p>Exactly one of ResourceClaimName and ResourceClaimTemplateName must be set.</p>

3.1.241. .template.spec.schedulingGates

Description

SchedulingGates is an opaque list of values that if specified will block scheduling the pod. If schedulingGates is not empty, the pod will stay in the SchedulingGated state and the scheduler will not attempt to schedule the pod.

SchedulingGates can only be set at pod creation time, and be removed only afterwards.

Type

array

3.1.242. .template.spec.schedulingGates[]

Description

PodSchedulingGate is associated to a Pod to guard its scheduling.

Type

object

Required

- **name**

Property	Type	Description
name	string	Name of the scheduling gate. Each scheduling gate must have a unique name field.

3.1.243. .template.spec.securityContext

Description

PodSecurityContext holds pod-level security attributes and common container settings. Some fields are also present in container.securityContext. Field values of container.securityContext take precedence over field values of PodSecurityContext.

Type

object

Property	Type	Description
appArmorProfile	object	AppArmorProfile defines a pod or container's AppArmor settings.
fsGroup	integer	<p>A special supplemental group that applies to all containers in a pod. Some volume types allow the Kubelet to change the ownership of that volume to be owned by the pod:</p> <ol style="list-style-type: none"> 1. The owning GID will be the FSGroup 2. The setgid bit is set (new files created in the volume will be owned by FSGroup) 3. The permission bits are OR'd with rw-rw---- <p>If unset, the Kubelet will not modify the ownership and permissions of any volume. Note that this field cannot be set when spec.os.name is windows.</p>

Property	Type	Description
fsGroupChangePolicy	string	<p>fsGroupChangePolicy defines behavior of changing ownership and permission of the volume before being exposed inside Pod. This field will only apply to volume types which support fsGroup based ownership (and permissions). It will have no effect on ephemeral volume types such as: secret, configmaps and emptydir. Valid values are "OnRootMismatch" and "Always". If not specified, "Always" is used. Note that this field cannot be set when spec.os.name is windows.</p> <p>Possible enum values: - "Always" indicates that volume's ownership and permissions should always be changed whenever volume is mounted inside a Pod. This the default behavior. - "OnRootMismatch" indicates that volume's ownership and permissions will be changed only when permission and ownership of root directory does not match with expected permissions on the volume. This can help shorten the time it takes to change ownership and permissions of a volume.</p>
runAsGroup	integer	<p>The GID to run the entrypoint of the container process. Uses runtime default if unset. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.</p>

Property	Type	Description
runAsNonRoot	boolean	Indicates that the container must run as a non-root user. If true, the Kubelet will validate the image at runtime to ensure that it does not run as UID 0 (root) and fail to start the container if it does. If unset or false, no such validation will be performed. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.
runAsUser	integer	The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified. May also be set in SecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence for that container. Note that this field cannot be set when spec.os.name is windows.
seLinuxOptions	object	SELinuxOptions are the labels to be applied to the container
seccompProfile	object	SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

Property	Type	Description
supplementalGroups	array (integer)	<p>A list of groups applied to the first process run in each container, in addition to the container's primary GID and fsGroup (if specified). If the SupplementalGroupsPolicy feature is enabled, the supplementalGroupsPolicy field determines whether these are in addition to or instead of any group memberships defined in the container image. If unspecified, no additional groups are added, though group memberships defined in the container image may still be used, depending on the supplementalGroupsPolicy field. Note that this field cannot be set when spec.os.name is windows.</p>
supplementalGroupsPolicy	string	<p>Defines how supplemental groups of the first container processes are calculated. Valid values are "Merge" and "Strict". If not specified, "Merge" is used. (Alpha) Using the field requires the SupplementalGroupsPolicy feature gate to be enabled and the container runtime must implement support for this feature. Note that this field cannot be set when spec.os.name is windows.</p> <p>Possible enum values: - "Merge" means that the container's provided SupplementalGroups and FsGroup (specified in SecurityContext) will be merged with the primary user's groups as defined in the container image (in /etc/group). - "Strict" means that the container's provided SupplementalGroups and FsGroup (specified in SecurityContext) will be used instead of any groups defined in the container image.</p>

Property	Type	Description
sysctls	array	Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.
sysctls[]	object	Sysctl defines a kernel parameter to be set
windowsOptions	object	WindowsSecurityContextOptions contain Windows-specific options and credentials.

3.1.244. .template.spec.securityContext.appArmorProfile

Description

AppArmorProfile defines a pod or container's AppArmor settings.

Type

object

Required

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile loaded on the node that should be used. The profile must be preconfigured on the node to work. Must match the loaded name of the profile. Must be set if and only if type is "Localhost".

Property	Type	Description
type	string	<p>type indicates which kind of AppArmor profile will be applied. Valid options are: Localhost - a profile pre-loaded on the node. RuntimeDefault - the container runtime's default profile. Unconfined - no AppArmor enforcement.</p> <p>Possible enum values: - "Localhost" indicates that a profile pre-loaded on the node should be used. - "RuntimeDefault" indicates that the container runtime's default AppArmor profile should be used. - "Unconfined" indicates that no AppArmor profile should be enforced.</p>

3.1.245. .template.spec.securityContext.seLinuxOptions

Description

SELinuxOptions are the labels to be applied to the container

Type

object

Property	Type	Description
level	string	Level is SELinux level label that applies to the container.
role	string	Role is a SELinux role label that applies to the container.
type	string	Type is a SELinux type label that applies to the container.
user	string	User is a SELinux user label that applies to the container.

3.1.246. .template.spec.securityContext.seccompProfile

Description

SeccompProfile defines a pod/container's seccomp profile settings. Only one profile source may be set.

Type

object**Required**

- **type**

Property	Type	Description
localhostProfile	string	localhostProfile indicates a profile defined in a file on the node should be used. The profile must be preconfigured on the node to work. Must be a descending path, relative to the kubelet's configured seccomp profile location. Must be set if type is "Localhost". Must NOT be set for any other type.
type	string	<p>type indicates which kind of seccomp profile will be applied. Valid options are:</p> <p>Localhost - a profile defined in a file on the node should be used. RuntimeDefault - the container runtime default profile should be used. Unconfined - no profile should be applied.</p> <p>Possible enum values: - "Localhost" indicates a profile defined in a file on the node should be used. The file's location relative to <kubelet-root-dir>/seccomp. - "RuntimeDefault" represents the default container runtime seccomp profile. - "Unconfined" indicates no seccomp profile is applied (A.K.A. unconfined).</p>

3.1.247. .template.spec.securityContext.sysctls**Description**

Sysctls hold a list of namespaced sysctls used for the pod. Pods with unsupported sysctls (by the container runtime) might fail to launch. Note that this field cannot be set when spec.os.name is windows.

Type

array

3.1.248. .template.spec.securityContext.sysctls[]

Description

Sysctl defines a kernel parameter to be set

Type

object

Required

- **name**
- **value**

Property	Type	Description
name	string	Name of a property to set
value	string	Value of a property to set

3.1.249. .template.spec.securityContext.windowsOptions

Description

WindowsSecurityContextOptions contain Windows-specific options and credentials.

Type

object

Property	Type	Description
gmsaCredentialSpec	string	GMSACredentialSpec is where the GMSA admission webhook (https://github.com/kubernetes-sigs/windows-gmsa) inlines the contents of the GMSA credential spec named by the <code>GMSACredentialSpecName</code> field.
gmsaCredentialSpecName	string	GMSACredentialSpecName is the name of the GMSA credential spec to use.

Property	Type	Description
hostProcess	boolean	HostProcess determines if a container should be run as a 'Host Process' container. All of a Pod's containers must have the same effective HostProcess value (it is not allowed to have a mix of HostProcess containers and non-HostProcess containers). In addition, if HostProcess is true then HostNetwork must also be set to true.
runAsUserName	string	The UserName in Windows to run the entrypoint of the container process. Defaults to the user specified in image metadata if unspecified. May also be set in PodSecurityContext. If set in both SecurityContext and PodSecurityContext, the value specified in SecurityContext takes precedence.

3.1.250. .template.spec.tolerations

Description

If specified, the pod's tolerations.

Type

array

3.1.251. .template.spec.tolerations[]

Description

The pod this Toleration is attached to tolerates any taint that matches the triple <key,value,effect> using the matching operator <operator>.

Type

object

Property	Type	Description
----------	------	-------------

Property	Type	Description
effect	string	<p>Effect indicates the taint effect to match. Empty means match all taint effects. When specified, allowed values are NoSchedule, PreferNoSchedule and NoExecute.</p> <p>Possible enum values: - "NoExecute" Evict any already-running pods that do not tolerate the taint. Currently enforced by NodeController. - "NoSchedule" Do not allow new pods to schedule onto the node unless they tolerate the taint, but allow all pods submitted to Kubelet without going through the scheduler to start, and allow all already-running pods to continue running. Enforced by the scheduler. - "PreferNoSchedule" Like TaintEffectNoSchedule, but the scheduler tries not to schedule new pods onto the node, rather than prohibiting new pods from scheduling onto the node entirely. Enforced by the scheduler.</p>
key	string	<p>Key is the taint key that the toleration applies to. Empty means match all taint keys. If the key is empty, operator must be Exists; this combination means to match all values and all keys.</p>
operator	string	<p>Operator represents a key's relationship to the value. Valid operators are Exists and Equal. Defaults to Equal. Exists is equivalent to wildcard for value, so that a pod can tolerate all taints of a particular category.</p> <p>Possible enum values: - "Equal" - "Exists"</p>

Property	Type	Description
tolerationSeconds	integer	TolerationSeconds represents the period of time the toleration (which must be of effect NoExecute, otherwise this field is ignored) tolerates the taint. By default, it is not set, which means tolerate the taint forever (do not evict). Zero and negative values will be treated as 0 (evict immediately) by the system.
value	string	Value is the taint value the toleration matches to. If the operator is Exists, the value should be empty, otherwise just a regular string.

3.1.252. .template.spec.topologySpreadConstraints

Description

TopologySpreadConstraints describes how a group of pods ought to spread across topology domains. Scheduler will schedule pods in a way which abides by the constraints. All topologySpreadConstraints are ANDed.

Type

array

3.1.253. .template.spec.topologySpreadConstraints[]

Description

TopologySpreadConstraint specifies how to spread matching pods among the given topology.

Type

object

Required

- **maxSkew**
- **topologyKey**
- **whenUnsatisfiable**

Property	Type	Description
----------	------	-------------

Property	Type	Description
labelSelector	LabelSelector	LabelSelector is used to find matching pods. Pods that match this label selector are counted to determine the number of pods in their corresponding topology domain.
matchLabelKeys	array (string)	<p>MatchLabelKeys is a set of pod label keys to select the pods over which spreading will be calculated. The keys are used to lookup values from the incoming pod labels, those key-value labels are ANDed with labelSelector to select the group of existing pods over which spreading will be calculated for the incoming pod. The same key is forbidden to exist in both MatchLabelKeys and LabelSelector. MatchLabelKeys cannot be set when LabelSelector isn't set. Keys that don't exist in the incoming pod labels will be ignored. A null or empty list means only match against labelSelector.</p> <p>This is a beta field and requires the MatchLabelKeysInPodTopologySpread feature gate to be enabled (enabled by default).</p>

Property	Type	Description
maxSkew	integer	<p>MaxSkew describes the degree to which pods may be unevenly distributed. When whenUnsatisfiable=DoNotSchedule, it is the maximum permitted difference between the number of matching pods in the target topology and the global minimum. The global minimum is the minimum number of matching pods in an eligible domain or zero if the number of eligible domains is less than MinDomains. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same labelSelector spread as 2/2/1: In this case, the global minimum is 1. zone1 zone2 zone3 P P P P P - if MaxSkew is 1, incoming pod can only be scheduled to zone3 to become 2/2/2; scheduling it onto zone1(zone2) would make the ActualSkew(3-1) on zone1(zone2) violate MaxSkew(1). - if MaxSkew is 2, incoming pod can be scheduled onto any zone. When whenUnsatisfiable=ScheduleAnyway, it is used to give higher precedence to topologies that satisfy it. It's a required field. Default value is 1 and 0 is not allowed.</p>

Property	Type	Description
minDomains	integer	<p>MinDomains indicates a minimum number of eligible domains. When the number of eligible domains with matching topology keys is less than minDomains, Pod Topology Spread treats "global minimum" as 0, and then the calculation of Skew is performed. And when the number of eligible domains with matching topology keys equals or greater than minDomains, this value has no effect on scheduling. As a result, when the number of eligible domains is less than minDomains, scheduler won't schedule more than maxSkew Pods to those domains. If value is nil, the constraint behaves as if MinDomains is equal to 1. Valid values are integers greater than 0. When value is not nil, WhenUnsatisfiable must be DoNotSchedule.</p> <p>For example, in a 3-zone cluster, MaxSkew is set to 2, MinDomains is set to 5 and pods with the same labelSelector spread as 2/2/2: zone1 zone2 zone3 P P P P P P The number of domains is less than 5(MinDomains), so "global minimum" is treated as 0. In this situation, new pod with the same labelSelector cannot be scheduled, because computed skew will be 3(3 - 0) if new Pod is scheduled to any of the three zones, it will violate MaxSkew.</p>

Property	Type	Description
nodeAffinityPolicy	string	<p>NodeAffinityPolicy indicates how we will treat Pod's nodeAffinity/nodeSelector when calculating pod topology spread skew. Options are: - Honor: only nodes matching nodeAffinity/nodeSelector are included in the calculations. - Ignore: nodeAffinity/nodeSelector are ignored. All nodes are included in the calculations.</p> <p>If this value is nil, the behavior is equivalent to the Honor policy. This is a beta-level feature default enabled by the NodeInclusionPolicyInPodTopologySpread feature flag.</p> <p>Possible enum values: - "Honor" means use this scheduling directive when calculating pod topology spread skew. - "Ignore" means ignore this scheduling directive when calculating pod topology spread skew.</p>

Property	Type	Description
nodeTaintsPolicy	string	<p>NodeTaintsPolicy indicates how we will treat node taints when calculating pod topology spread skew. Options are: - Honor: nodes without taints, along with tainted nodes for which the incoming pod has a toleration, are included. - Ignore: node taints are ignored. All nodes are included.</p> <p>If this value is nil, the behavior is equivalent to the Ignore policy. This is a beta-level feature default enabled by the NodeInclusionPolicyInPodTopologySpread feature flag.</p> <p>Possible enum values: - "Honor" means use this scheduling directive when calculating pod topology spread skew. - "Ignore" means ignore this scheduling directive when calculating pod topology spread skew.</p>
topologyKey	string	<p>TopologyKey is the key of node labels. Nodes that have a label with this key and identical values are considered to be in the same topology. We consider each <key, value> as a "bucket", and try to put balanced number of pods into each bucket. We define a domain as a particular instance of a topology. Also, we define an eligible domain as a domain whose nodes meet the requirements of nodeAffinityPolicy and nodeTaintsPolicy. e.g. If TopologyKey is "kubernetes.io/hostname", each Node is a domain of that topology. And, if TopologyKey is "topology.kubernetes.io/zone", each zone is a domain of that topology. It's a required field.</p>

Property	Type	Description
whenUnsatisfiable	string	<p>WhenUnsatisfiable indicates how to deal with a pod if it doesn't satisfy the spread constraint. - DoNotSchedule (default) tells the scheduler not to schedule it. - ScheduleAnyway tells the scheduler to schedule the pod in any location, but giving higher precedence to topologies that would help reduce the skew. A constraint is considered "Unsatisfiable" for an incoming pod if and only if every possible node assignment for that pod would violate "MaxSkew" on some topology. For example, in a 3-zone cluster, MaxSkew is set to 1, and pods with the same labelSelector spread as 3/1/1: zone1 zone2 zone3 P P P P P If WhenUnsatisfiable is set to DoNotSchedule, incoming pod can only be scheduled to zone2(zone3) to become 3/2/1(3/1/2) as ActualSkew(2-1) on zone2(zone3) satisfies MaxSkew(1). In other words, the cluster can still be imbalanced, but scheduler won't make it more imbalanced. It's a required field.</p> <p>Possible enum values: - "DoNotSchedule" instructs the scheduler not to schedule the pod when constraints are not satisfied. - "ScheduleAnyway" instructs the scheduler to schedule the pod even if constraints are not satisfied.</p>

3.1.254. .template.spec.volumes

Description

List of volumes that can be mounted by containers belonging to the pod. More info: <https://kubernetes.io/docs/concepts/storage/volumes>

Type

array

3.1.255. .template.spec.volumes[]

Description

Volume represents a named volume in a pod that may be accessed by any container in the pod.

Type

object

Required

- **name**

Property	Type	Description
awsElasticBlockStore	object	<p>Represents a Persistent Disk resource in AWS.</p> <p>An AWS EBS disk must exist before mounting to a container. The disk must also be in the same AWS zone as the kubelet. An AWS EBS disk can only be mounted as read/write once. AWS EBS volumes support ownership management and SELinux relabeling.</p>
azureDisk	object	AzureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.
azureFile	object	AzureFile represents an Azure File Service mount on the host and bind mount to the pod.
cephfs	object	Represents a Ceph Filesystem mount that lasts the lifetime of a pod Cephfs volumes do not support ownership management or SELinux relabeling.
cinder	object	Represents a cinder volume resource in Openstack. A Cinder volume must exist before mounting to a container. The volume must also be in the same region as the kubelet. Cinder volumes support ownership management and SELinux relabeling.

Property	Type	Description
configMap	object	<p>Adapts a ConfigMap into a volume.</p> <p>The contents of the target ConfigMap's Data field will be presented in a volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. ConfigMap volumes support ownership management and SELinux relabeling.</p>
csi	object	Represents a source location of a volume to mount, managed by an external CSI driver
downwardAPI	object	DownwardAPIVolumeSource represents a volume containing downward API info. Downward API volumes support ownership management and SELinux relabeling.
emptyDir	object	Represents an empty directory for a pod. Empty directory volumes support ownership management and SELinux relabeling.
ephemeral	object	Represents an ephemeral volume that is handled by a normal storage driver.
fc	object	Represents a Fibre Channel volume. Fibre Channel volumes can only be mounted as read/write once. Fibre Channel volumes support ownership management and SELinux relabeling.
flexVolume	object	FlexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.

Property	Type	Description
flocker	object	Represents a Flocker volume mounted by the Flocker agent. One and only one of datasetName and datasetUUID should be set. Flocker volumes do not support ownership management or SELinux relabeling.
gcePersistentDisk	object	Represents a Persistent Disk resource in Google Compute Engine. A GCE PD must exist before mounting to a container. The disk must also be in the same GCE project and zone as the kubelet. A GCE PD can only be mounted as read/write once or read-only many times. GCE PDs support ownership management and SELinux relabeling.
gitRepo	object	Represents a volume that is populated with the contents of a git repository. Git repo volumes do not support ownership management. Git repo volumes support SELinux relabeling. DEPRECATED: GitRepo is deprecated. To provision a container with a git repo, mount an EmptyDir into an InitContainer that clones the repo using git, then mount the EmptyDir into the Pod's container.
glusterfs	object	Represents a Glusterfs mount that lasts the lifetime of a pod. Glusterfs volumes do not support ownership management or SELinux relabeling.
hostPath	object	Represents a host path mapped into a pod. Host path volumes do not support ownership management or SELinux relabeling.
image	object	ImageVolumeSource represents a image volume resource.

Property	Type	Description
iscsi	object	Represents an iSCSI disk. iSCSI volumes can only be mounted as read/write once. iSCSI volumes support ownership management and SELinux relabeling.
name	string	name of the volume. Must be a DNS_LABEL and unique within the pod. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
nfs	object	Represents an NFS mount that lasts the lifetime of a pod. NFS volumes do not support ownership management or SELinux relabeling.
persistentVolumeClaim	object	PersistentVolumeClaimVolumeSource references the user's PVC in the same namespace. This volume finds the bound PV and mounts that volume for the pod. A PersistentVolumeClaimVolumeSource is, essentially, a wrapper around another type of volume that is owned by someone else (the system).
photonPersistentDisk	object	Represents a Photon Controller persistent disk resource.
portworxVolume	object	PortworxVolumeSource represents a Portworx volume resource.
projected	object	Represents a projected volume source
quobyte	object	Represents a Quobyte mount that lasts the lifetime of a pod. Quobyte volumes do not support ownership management or SELinux relabeling.

Property	Type	Description
rbd	object	Represents a Rados Block Device mount that lasts the lifetime of a pod. RBD volumes support ownership management and SELinux relabeling.
scaleIO	object	ScaleIOVolumeSource represents a persistent ScaleIO volume
secret	object	Adapts a Secret into a volume. The contents of the target Secret's Data field will be presented in a volume as files using the keys in the Data field as the file names. Secret volumes support ownership management and SELinux relabeling.
storageos	object	Represents a StorageOS persistent volume resource.
vsphereVolume	object	Represents a vSphere volume resource.

3.1.256. .template.spec.volumes[].awsElasticBlockStore

Description

Represents a Persistent Disk resource in AWS.

An AWS EBS disk must exist before mounting to a container. The disk must also be in the same AWS zone as the kubelet. An AWS EBS disk can only be mounted as read/write once. AWS EBS volumes support ownership management and SELinux relabeling.

Type

object

Required

- **volumeID**

Property	Type	Description
----------	------	-------------

Property	Type	Description
fsType	string	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
partition	integer	partition is the partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty).
readOnly	boolean	readOnly value true will force the readOnly setting in VolumeMounts. More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore
volumeID	string	volumeID is unique ID of the persistent disk resource in AWS (Amazon EBS volume). More info: https://kubernetes.io/docs/concepts/storage/volumes#awselasticblockstore

3.1.257. .template.spec.volumes[].azureDisk

Description

AzureDisk represents an Azure Data Disk mount on the host and bind mount to the pod.

Type

object

Required

- **diskName**
- **diskURI**

Property	Type	Description
cachingMode	string	<p>cachingMode is the Host Caching mode: None, Read Only, Read Write.</p> <p>Possible enum values: - "None" - "ReadOnly" - "ReadWrite"</p>
diskName	string	diskName is the Name of the data disk in the blob storage
diskURI	string	diskURI is the URI of data disk in the blob storage
fsType	string	<p>fsType is Filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.</p>
kind	string	<p>kind expected values are Shared: multiple blob disks per storage account Dedicated: single blob disk per storage account Managed: azure managed data disk (only in managed availability set). defaults to shared</p> <p>Possible enum values: - "Dedicated" - "Managed" - "Shared"</p>
readOnly	boolean	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

3.1.258. .template.spec.volumes[].azureFile

Description

AzureFile represents an Azure File Service mount on the host and bind mount to the pod.

Type

object

Required

- **secretName**
- **shareName**

Property	Type	Description
readOnly	boolean	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
secretName	string	secretName is the name of secret that contains Azure Storage Account Name and Key
shareName	string	shareName is the azure share Name

3.1.259. .template.spec.volumes[].cephfs

Description

Represents a Ceph Filesystem mount that lasts the lifetime of a pod Cephfs volumes do not support ownership management or SELinux relabeling.

Type

object

Required

- **monitors**

Property	Type	Description
monitors	array (string)	monitors is Required: Monitors is a collection of Ceph monitors More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
path	string	path is Optional: Used as the mounted root, rather than the full Ceph tree, default is /
readOnly	boolean	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

Property	Type	Description
secretFile	string	secretFile is Optional: SecretFile is the path to key ring for User, default is /etc/ceph/user.secret More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
user	string	user is optional: User is the rados user name, default is admin More info: https://examples.k8s.io/volumes/cephfs/README.md#how-to-use-it

3.1.260. .template.spec.volumes[].cephfs.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

3.1.261. .template.spec.volumes[].cinder

Description

Represents a cinder volume resource in Openstack. A Cinder volume must exist before mounting to a container. The volume must also be in the same region as the kubelet. Cinder volumes support ownership management and SELinux relabeling.

Type

object**Required**

- **volumeID**

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://examples.k8s.io/mysql-cinder-pd/README.md
readOnly	boolean	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts. More info: https://examples.k8s.io/mysql-cinder-pd/README.md
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
volumeID	string	volumeID used to identify the volume in cinder. More info: https://examples.k8s.io/mysql-cinder-pd/README.md

3.1.262. .template.spec.volumes[].cinder.secretRef**Description**

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type**object**

Property	Type	Description
----------	------	-------------

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

3.1.263. .template.spec.volumes[].configMap

Description

Adapts a ConfigMap into a volume.

The contents of the target ConfigMap's Data field will be presented in a volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. ConfigMap volumes support ownership management and SELinux relabeling.

Type

object

Property	Type	Description
defaultMode	integer	defaultMode is optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Property	Type	Description
items	array	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
items[]	object	Maps a string key to a path within a volume.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	optional specify whether the ConfigMap or its keys must be defined

3.1.264. .template.spec.volumes[].configMap.items

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

3.1.265. .template.spec.volumes[].configMap.items[]

Description

Maps a string key to a path within a volume.

Type

object

Required

- **key**
- **path**

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

3.1.266. .template.spec.volumes[].csi

Description

Represents a source location of a volume to mount, managed by an external CSI driver

Type

object

Required

- **driver**

Property	Type	Description
----------	------	-------------

Property	Type	Description
driver	string	driver is the name of the CSI driver that handles this volume. Consult with your admin for the correct name as registered in the cluster.
fsType	string	fsType to mount. Ex. "ext4", "xfs", "ntfs". If not provided, the empty value is passed to the associated CSI driver which will determine the default filesystem to apply.
nodePublishSecretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
readOnly	boolean	readOnly specifies a read-only configuration for the volume. Defaults to false (read/write).
volumeAttributes	object (string)	volumeAttributes stores driver-specific properties that are passed to the CSI driver. Consult your driver's documentation for supported values.

3.1.267. .template.spec.volumes[].csi.nodePublishSecretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

3.1.268. .template.spec.volumes[].downwardAPI

Description

DownwardAPIVolumeSource represents a volume containing downward API info. Downward API volumes support ownership management and SELinux relabeling.

Type

object

Property	Type	Description
defaultMode	integer	Optional: mode bits to use on created files by default. Must be a Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
items	array	Items is a list of downward API volume file
items[]	object	DownwardAPIVolumeFile represents information to create the file containing the pod field

3.1.269. .template.spec.volumes[].downwardAPI.items

Description

Items is a list of downward API volume file

Type

array

3.1.270. .template.spec.volumes[].downwardAPI.items[]

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

object

Required

- **path**

Property	Type	Description
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.
mode	integer	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
resourceFieldRef	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format

3.1.271. .template.spec.volumes[].downwardAPI.items[].fieldRef

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

- **fieldPath**

Property	Type	Description
apiVersion	string	Version of the schema the FieldPath is written in terms of, defaults to "v1".

Property	Type	Description
fieldPath	string	Path of the field to select in the specified API version.

3.1.272. .template.spec.volumes[].downwardAPI.items[].resourceFieldRef

Description

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

- **resource**

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	Quantity	Specifies the output format of the exposed resources, defaults to "1"
resource	string	Required: resource to select

3.1.273. .template.spec.volumes[].emptyDir

Description

Represents an empty directory for a pod. Empty directory volumes support ownership management and SELinux relabeling.

Type

object

Property	Type	Description
medium	string	medium represents what type of storage medium should back this directory. The default is "" which means to use the node's default medium. Must be an empty string (default) or Memory. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

Property	Type	Description
sizeLimit	Quantity	sizeLimit is the total amount of local storage required for this EmptyDir volume. The size limit is also applicable for memory medium. The maximum usage on memory medium EmptyDir would be the minimum value between the SizeLimit specified here and the sum of memory limits of all containers in a pod. The default is nil which means that the limit is undefined. More info: https://kubernetes.io/docs/concepts/storage/volumes#emptydir

3.1.274. .template.spec.volumes[].ephemeral

Description

Represents an ephemeral volume that is handled by a normal storage driver.

Type

object

Property	Type	Description
volumeClaimTemplate	object	PersistentVolumeClaimTemplate is used to produce PersistentVolumeClaim objects as part of an EphemeralVolumeSource.

3.1.275. .template.spec.volumes[].ephemeral.volumeClaimTemplate

Description

PersistentVolumeClaimTemplate is used to produce PersistentVolumeClaim objects as part of an EphemeralVolumeSource.

Type

object

Required

- **spec**

Property	Type	Description
metadata	ObjectMeta	May contain labels and annotations that will be copied into the PVC when creating it. No other fields are allowed and will be rejected during validation.
spec	object	PersistentVolumeClaimSpec describes the common attributes of storage devices and allows a Source for provider-specific attributes

3.1.276. .template.spec.volumes[].ephemeral.volumeClaimTemplate.spec

Description

PersistentVolumeClaimSpec describes the common attributes of storage devices and allows a Source for provider-specific attributes

Type

object

Property	Type	Description
accessModes	array (string)	accessModes contains the desired access modes the volume should have. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#access-modes-1
dataSource	object	TypedLocalObjectReference contains enough information to let you locate the typed referenced object inside the same namespace.

Property	Type	Description
dataSourceRef	object	<p>dataSourceRef specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a PersistentVolumeClaim object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the dataSource field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in dataSourceRef, both fields (dataSource and dataSourceRef) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in dataSourceRef, dataSource isn't set to the same value and must be empty. There are three important differences between dataSource and dataSourceRef:</p> <ul style="list-style-type: none"> * While dataSource only allows two specific types of objects, dataSourceRef allows any non-core object, as well as PersistentVolumeClaim objects. * While dataSource ignores disallowed values (dropping them), dataSourceRef preserves all values, and generates an error if a disallowed value is specified. * While dataSource only allows local objects, dataSourceRef allows objects in any namespaces. <p>(Beta) Using this field requires the AnyVolumeDataSource feature gate to be enabled.</p> <p>(Alpha) Using the namespace field of dataSourceRef requires the CrossNamespaceVolumeDataSource feature gate to be enabled.</p>

Property	Type	Description
resources	object	VolumeResourceRequirements describes the storage resource requirements for a volume.
selector	LabelSelector	selector is a label query over volumes to consider for binding.
storageClassName	string	storageClassName is the name of the StorageClass required by the claim. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#class-1

Property	Type	Description
volumeAttributesClassName	string	<p>volumeAttributesClassName may be used to set the VolumeAttributesClass used by this claim. If specified, the CSI driver will create or update the volume with the attributes defined in the corresponding VolumeAttributesClass. This has a different purpose than storageClassName, it can be changed after the claim is created. An empty string value means that no VolumeAttributesClass will be applied to the claim but it's not allowed to reset this field to empty string once it is set. If unspecified and the PersistentVolumeClaim is unbound, the default VolumeAttributesClass will be set by the persistentvolume controller if it exists. If the resource referred to by volumeAttributesClass does not exist, this PersistentVolumeClaim will be set to a Pending state, as reflected by the modifyVolumeStatus field, until such as a resource exists. More info: https://kubernetes.io/docs/concepts/storage/volume-attributes-classes/ (Beta) Using this field requires the VolumeAttributesClass feature gate to be enabled (off by default).</p>

Property	Type	Description
volumeMode	string	<p>volumeMode defines what type of volume is required by the claim. Value of Filesystem is implied when not included in claim spec.</p> <p>Possible enum values: - "Block" means the volume will not be formatted with a filesystem and will remain a raw block device. - "Filesystem" means the volume will be or is formatted with a filesystem.</p>
volumeName	string	<p>volumeName is the binding reference to the PersistentVolume backing this claim.</p>

3.1.277. .template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSource

Description

TypedLocalObjectReference contains enough information to let you locate the typed referenced object inside the same namespace.

Type

object

Required

- **kind**
- **name**

Property	Type	Description
apiGroup	string	<p>APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.</p>
kind	string	<p>Kind is the type of resource being referenced</p>
name	string	<p>Name is the name of resource being referenced</p>

3.1.278. .template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.dataSourceF

Description

`dataSourceRef` specifies the object from which to populate the volume with data, if a non-empty volume is desired. This may be any object from a non-empty API group (non core object) or a `PersistentVolumeClaim` object. When this field is specified, volume binding will only succeed if the type of the specified object matches some installed volume populator or dynamic provisioner. This field will replace the functionality of the `dataSource` field and as such if both fields are non-empty, they must have the same value. For backwards compatibility, when namespace isn't specified in `dataSourceRef`, both fields (`dataSource` and `dataSourceRef`) will be set to the same value automatically if one of them is empty and the other is non-empty. When namespace is specified in `dataSourceRef`, `dataSource` isn't set to the same value and must be empty. There are three important differences between `dataSource` and `dataSourceRef`: * While `dataSource` only allows two specific types of objects, `dataSourceRef` allows any non-core object, as well as `PersistentVolumeClaim` objects. * While `dataSource` ignores disallowed values (dropping them), `dataSourceRef` preserves all values, and generates an error if a disallowed value is specified. * While `dataSource` only allows local objects, `dataSourceRef` allows objects in any namespaces. (Beta) Using this field requires the `AnyVolumeDataSource` feature gate to be enabled. (Alpha) Using the namespace field of `dataSourceRef` requires the `CrossNamespaceVolumeDataSource` feature gate to be enabled.

Type

object

Required

- **kind**
- **name**

Property	Type	Description
apiGroup	string	APIGroup is the group for the resource being referenced. If APIGroup is not specified, the specified Kind must be in the core API group. For any other third-party types, APIGroup is required.
kind	string	Kind is the type of resource being referenced
name	string	Name is the name of resource being referenced

Property	Type	Description
namespace	string	Namespace is the namespace of resource being referenced Note that when a namespace is specified, a gateway.networking.k8s.io/ReferenceGrant object is required in the referent namespace to allow that namespace's owner to accept the reference. See the ReferenceGrant documentation for details. (Alpha) This field requires the CrossNamespaceVolumeDataSource feature gate to be enabled.

3.1.279. .template.spec.volumes[].ephemeral.volumeClaimTemplate.spec.resources

Description

VolumeResourceRequirements describes the storage resource requirements for a volume.

Type

object

Property	Type	Description
limits	object (Quantity)	Limits describes the maximum amount of compute resources allowed. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
requests	object (Quantity)	Requests describes the minimum amount of compute resources required. If Requests is omitted for a container, it defaults to Limits if that is explicitly specified, otherwise to an implementation-defined value. Requests cannot exceed Limits. More info: https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

3.1.280. .template.spec.volumes[].fc

Description

Represents a Fibre Channel volume. Fibre Channel volumes can only be mounted as read/write once. Fibre Channel volumes support ownership management and SELinux relabeling.

Type
object

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
lun	integer	lun is Optional: FC target lun number
readOnly	boolean	readOnly is Optional: Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
targetWWNs	array (string)	targetWWNs is Optional: FC target worldwide names (WWNs)
wwids	array (string)	wwids Optional: FC volume world wide identifiers (wwids) Either wwids or combination of targetWWNs and lun must be set, but not both simultaneously.

3.1.281. .template.spec.volumes[].flexVolume

Description

FlexVolume represents a generic volume resource that is provisioned/attached using an exec based plugin.

Type
object

Required

- **driver**

Property	Type	Description
driver	string	driver is the name of the driver to use for this volume.

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". The default filesystem depends on FlexVolume script.
options	object (string)	options is Optional: this field holds extra command options if any.
readOnly	boolean	readOnly is Optional: defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

3.1.282. .template.spec.volumes[].flexVolume.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

3.1.283. .template.spec.volumes[].flocker

Description

Represents a Flocker volume mounted by the Flocker agent. One and only one of datasetName and datasetUUID should be set. Flocker volumes do not support ownership management or SELinux relabeling.

Type

object

Property	Type	Description
datasetName	string	datasetName is Name of the dataset stored as metadata → name on the dataset for Flocker should be considered as deprecated
datasetUUID	string	datasetUUID is the UUID of the dataset. This is unique identifier of a Flocker dataset

3.1.284. .template.spec.volumes[].gcePersistentDisk

Description

Represents a Persistent Disk resource in Google Compute Engine.

A GCE PD must exist before mounting to a container. The disk must also be in the same GCE project and zone as the kubelet. A GCE PD can only be mounted as read/write once or read-only many times. GCE PDs support ownership management and SELinux relabeling.

Type

object

Required

- **pdName**

Property	Type	Description
fsType	string	fsType is filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

Property	Type	Description
partition	integer	partition is the partition in the volume that you want to mount. If omitted, the default is to mount by volume name. Examples: For volume /dev/sda1, you specify the partition as "1". Similarly, the volume partition for /dev/sda is "0" (or you can leave the property empty). More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
pdName	string	pdName is unique name of the PD resource in GCE. Used to identify the disk in GCE. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk
readOnly	boolean	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#gcepersistentdisk

3.1.285. .template.spec.volumes[].gitRepo

Description

Represents a volume that is populated with the contents of a git repository. Git repo volumes do not support ownership management. Git repo volumes support SELinux relabeling.

DEPRECATED: GitRepo is deprecated. To provision a container with a git repo, mount an EmptyDir into an InitContainer that clones the repo using git, then mount the EmptyDir into the Pod's container.

Type

object

Required

- **repository**

Property	Type	Description
----------	------	-------------

Property	Type	Description
directory	string	directory is the target directory name. Must not contain or start with '..'. If '.' is supplied, the volume directory will be the git repository. Otherwise, if specified, the volume will contain the git repository in the subdirectory with the given name.
repository	string	repository is the URL
revision	string	revision is the commit hash for the specified revision.

3.1.286. .template.spec.volumes[].glusterfs

Description

Represents a Glusterfs mount that lasts the lifetime of a pod. Glusterfs volumes do not support ownership management or SELinux relabeling.

Type

object

Required

- **endpoints**
- **path**

Property	Type	Description
endpoints	string	endpoints is the endpoint name that details Glusterfs topology. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod
path	string	path is the Glusterfs volume path. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

Property	Type	Description
readOnly	boolean	readOnly here will force the Glusterfs volume to be mounted with read-only permissions. Defaults to false. More info: https://examples.k8s.io/volumes/glusterfs/README.md#create-a-pod

3.1.287. .template.spec.volumes[].hostPath

Description

Represents a host path mapped into a pod. Host path volumes do not support ownership management or SELinux relabeling.

Type

object

Required

- **path**

Property	Type	Description
path	string	path of the directory on the host. If the path is a symlink, it will follow the link to the real path. More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath

Property	Type	Description
type	string	<p>type for HostPath Volume Defaults to "" More info: https://kubernetes.io/docs/concepts/storage/volumes#hostpath</p> <p>Possible enum values: - "" For backwards compatible, leave it empty if unset - "BlockDevice" A block device must exist at the given path - "CharDevice" A character device must exist at the given path - "Directory" A directory must exist at the given path - "DirectoryOrCreate" If nothing exists at the given path, an empty directory will be created there as needed with file mode 0755, having the same group and ownership with Kubelet. - "File" A file must exist at the given path - "FileOrCreate" If nothing exists at the given path, an empty file will be created there as needed with file mode 0644, having the same group and ownership with Kubelet. - "Socket" A UNIX socket must exist at the given path</p>

3.1.288. .template.spec.volumes[].image

Description

ImageVolumeSource represents a image volume resource.

Type

object

Property	Type	Description
----------	------	-------------

Property	Type	Description
pullPolicy	string	<p>Policy for pulling OCI objects. Possible values are: Always: the kubelet always attempts to pull the reference. Container creation will fail If the pull fails. Never: the kubelet never pulls the reference and only uses a local image or artifact. Container creation will fail if the reference isn't present. IfNotPresent: the kubelet pulls if the reference isn't already present on disk. Container creation will fail if the reference isn't present and the pull fails. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise.</p> <p>Possible enum values: - "Always" means that kubelet always attempts to pull the latest image. Container will fail If the pull fails. - "IfNotPresent" means that kubelet pulls if the image isn't present on disk. Container will fail if the image isn't present and the pull fails. - "Never" means that kubelet never pulls an image, but only uses a local image. Container will fail if the image isn't present</p>
reference	string	<p>Required: Image or artifact reference to be used. Behaves in the same way as pod.spec.containers[*].image. Pull secrets will be assembled in the same way as for the container image by looking up node credentials, SA image pull secrets, and pod spec image pull secrets. More info: https://kubernetes.io/docs/concepts/containers/images This field is optional to allow higher level config management to default or override container images in workload controllers like Deployments and StatefulSets.</p>

3.1.289. .template.spec.volumes[].iscsi

Description

Represents an iSCSI disk. iSCSI volumes can only be mounted as read/write once. iSCSI volumes support ownership management and SELinux relabeling.

Type

object

Required

- **targetPortal**
- **iqn**
- **lun**

Property	Type	Description
chapAuthDiscovery	boolean	chapAuthDiscovery defines whether support iSCSI Discovery CHAP authentication
chapAuthSession	boolean	chapAuthSession defines whether support iSCSI Session CHAP authentication
fsType	string	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#iscsi
initiatorName	string	initiatorName is the custom iSCSI Initiator Name. If initiatorName is specified with iscsiInterface simultaneously, new iSCSI interface <target portal>:<volume name> will be created for the connection.
iqn	string	iqn is the target iSCSI Qualified Name.
iscsiInterface	string	iscsiInterface is the interface Name that uses an iSCSI transport. Defaults to 'default' (tcp).

Property	Type	Description
lun	integer	lun represents iSCSI Target Lun number.
portals	array (string)	portals is the iSCSI Target Portal List. The portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).
readOnly	boolean	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false.
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
targetPortal	string	targetPortal is iSCSI Target Portal. The Portal is either an IP or ip_addr:port if the port is other than default (typically TCP ports 860 and 3260).

3.1.290. .template.spec.volumes[].iscsi.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

3.1.291. .template.spec.volumes[].nfs

Description

Represents an NFS mount that lasts the lifetime of a pod. NFS volumes do not support ownership management or SELinux relabeling.

Type

object

Required

- **server**
- **path**

Property	Type	Description
path	string	path that is exported by the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs
readOnly	boolean	readOnly here will force the NFS export to be mounted with read-only permissions. Defaults to false. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs
server	string	server is the hostname or IP address of the NFS server. More info: https://kubernetes.io/docs/concepts/storage/volumes#nfs

3.1.292. .template.spec.volumes[].persistentVolumeClaim**Description**

PersistentVolumeClaimVolumeSource references the user's PVC in the same namespace. This volume finds the bound PV and mounts that volume for the pod. A

PersistentVolumeClaimVolumeSource is, essentially, a wrapper around another type of volume that is owned by someone else (the system).

Type

object

Required

- **claimName**

Property	Type	Description
----------	------	-------------

Property	Type	Description
claimName	string	claimName is the name of a PersistentVolumeClaim in the same namespace as the pod using this volume. More info: https://kubernetes.io/docs/concepts/storage/persistent-volumes#persistentvolumeclaims
readOnly	boolean	readOnly Will force the ReadOnly setting in VolumeMounts. Default false.

3.1.293. .template.spec.volumes[].photonPersistentDisk

Description

Represents a Photon Controller persistent disk resource.

Type

object

Required

- **pdID**

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
pdID	string	pdID is the ID that identifies Photon Controller persistent disk

3.1.294. .template.spec.volumes[].portworxVolume

Description

PortworxVolumeSource represents a Portworx volume resource.

Type

object

Required

- **volumeID**

Property	Type	Description
fsType	string	fSType represents the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs". Implicitly inferred to be "ext4" if unspecified.
readOnly	boolean	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
volumeID	string	volumeID uniquely identifies a Portworx volume

3.1.295. .template.spec.volumes[].projected

Description

Represents a projected volume source

Type

object

Property	Type	Description
defaultMode	integer	defaultMode are the mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
sources	array	sources is the list of volume projections. Each entry in this list handles one source.
sources[]	object	Projection that may be projected along with other supported volume types. Exactly one of these fields must be set.

3.1.296. .template.spec.volumes[].projected.sources

Description

sources is the list of volume projections. Each entry in this list handles one source.

Type

array

3.1.297. .template.spec.volumes[].projected.sources[]

Description

Projection that may be projected along with other supported volume types. Exactly one of these fields must be set.

Type

object

Property	Type	Description
clusterTrustBundle	object	ClusterTrustBundleProjection describes how to select a set of ClusterTrustBundle objects and project their contents into the pod filesystem.
configMap	object	Adapts a ConfigMap into a projected volume. The contents of the target ConfigMap's Data field will be presented in a projected volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. Note that this is identical to a configmap volume source without the default mode.
downwardAPI	object	Represents downward API info for projecting into a projected volume. Note that this is identical to a downwardAPI volume source without the default mode.

Property	Type	Description
secret	object	<p>Adapts a secret into a projected volume.</p> <p>The contents of the target Secret's Data field will be presented in a projected volume as files using the keys in the Data field as the file names. Note that this is identical to a secret volume source without the default mode.</p>
serviceAccountToken	object	<p>ServiceAccountTokenProjection represents a projected service account token volume. This projection can be used to insert a service account token into the pods runtime filesystem for use against APIs (Kubernetes API Server or otherwise).</p>

3.1.298. .template.spec.volumes[].projected.sources[].clusterTrustBundle

Description

ClusterTrustBundleProjection describes how to select a set of ClusterTrustBundle objects and project their contents into the pod filesystem.

Type

object

Required

- **path**

Property	Type	Description
labelSelector	LabelSelector	<p>Select all ClusterTrustBundles that match this label selector. Only has effect if signerName is set. Mutually-exclusive with name. If unset, interpreted as "match nothing". If set but empty, interpreted as "match everything".</p>

Property	Type	Description
name	string	Select a single ClusterTrustBundle by object name. Mutually-exclusive with signerName and labelSelector.
optional	boolean	If true, don't block pod startup if the referenced ClusterTrustBundle(s) aren't available. If using name, then the named ClusterTrustBundle is allowed not to exist. If using signerName, then the combination of signerName and labelSelector is allowed to match zero ClusterTrustBundles.
path	string	Relative path from the volume root to write the bundle.
signerName	string	Select all ClusterTrustBundles that match this signer name. Mutually-exclusive with name. The contents of all selected ClusterTrustBundles will be unified and deduplicated.

3.1.299. .template.spec.volumes[].projected.sources[].configMap

Description

Adapts a ConfigMap into a projected volume.

The contents of the target ConfigMap's Data field will be presented in a projected volume as files using the keys in the Data field as the file names, unless the items element is populated with specific mappings of keys to paths. Note that this is identical to a configmap volume source without the default mode.

Type

object

Property	Type	Description
----------	------	-------------

Property	Type	Description
items	array	items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
items[]	object	Maps a string key to a path within a volume.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	optional specify whether the ConfigMap or its keys must be defined

3.1.300. .template.spec.volumes[].projected.sources[].configMap.items

Description

items if unspecified, each key-value pair in the Data field of the referenced ConfigMap will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the ConfigMap, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

3.1.301. .template.spec.volumes[].projected.sources[].configMap.items[]

Description

Maps a string key to a path within a volume.

Type

object

Required

- **key**
- **path**

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

3.1.302. .template.spec.volumes[].projected.sources[].downwardAPI

Description

Represents downward API info for projecting into a projected volume. Note that this is identical to a downwardAPI volume source without the default mode.

Type

object

Property	Type	Description
items	array	Items is a list of DownwardAPIVolume file
items[]	object	DownwardAPIVolumeFile represents information to create the file containing the pod field

3.1.303. `.template.spec.volumes[].projected.sources[].downwardAPI.items`

Description

Items is a list of DownwardAPIVolume file

Type

array

3.1.304. `.template.spec.volumes[].projected.sources[].downwardAPI.items[]`

Description

DownwardAPIVolumeFile represents information to create the file containing the pod field

Type

object

Required

- **path**

Property	Type	Description
fieldRef	object	ObjectFieldSelector selects an APIVersioned field of an object.
mode	integer	Optional: mode bits used to set permissions on this file, must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	Required: Path is the relative path name of the file to be created. Must not be absolute or contain the '..' path. Must be utf-8 encoded. The first item of the relative path must not start with '..'
resourceFieldRef	object	ResourceFieldSelector represents container resources (cpu, memory) and their output format

3.1.305. `.template.spec.volumes[].projected.sources[].downwardAPI.items[].fieldRef`

Description

ObjectFieldSelector selects an APIVersioned field of an object.

Type

object

Required

- **fieldPath**

Property	Type	Description
apiVersion	string	Version of the schema the FieldPath is written in terms of, defaults to "v1".
fieldPath	string	Path of the field to select in the specified API version.

3.1.306. .template.spec.volumes[].projected.sources[].downwardAPI.items[].resourceFieldSelector**Description**

ResourceFieldSelector represents container resources (cpu, memory) and their output format

Type

object

Required

- **resource**

Property	Type	Description
containerName	string	Container name: required for volumes, optional for env vars
divisor	Quantity	Specifies the output format of the exposed resources, defaults to "1"
resource	string	Required: resource to select

3.1.307. .template.spec.volumes[].projected.sources[].secret**Description**

Adapts a secret into a projected volume.

The contents of the target Secret's Data field will be presented in a projected volume as files using the keys in the Data field as the file names. Note that this is identical to a secret volume source without the default mode.

Type

object

Property	Type	Description
items	array	items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
items[]	object	Maps a string key to a path within a volume.
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names
optional	boolean	optional field specify whether the Secret or its key must be defined

3.1.308. `.template.spec.volumes[].projected.sources[].secret.items`

Description

items if unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

3.1.309. `.template.spec.volumes[].projected.sources[].secret.items[]`

Description

Maps a string key to a path within a volume.

Type

object

Required

- **key**
- **path**

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

3.1.310. .template.spec.volumes[].projected.sources[].serviceAccountToken

Description

ServiceAccountTokenProjection represents a projected service account token volume. This projection can be used to insert a service account token into the pods runtime filesystem for use against APIs (Kubernetes API Server or otherwise).

Type

object

Required

- **path**

Property	Type	Description
audience	string	audience is the intended audience of the token. A recipient of a token must identify itself with an identifier specified in the audience of the token, and otherwise should reject the token. The audience defaults to the identifier of the apiserver.
expirationSeconds	integer	expirationSeconds is the requested duration of validity of the service account token. As the token approaches expiration, the kubelet volume plugin will proactively rotate the service account token. The kubelet will start trying to rotate the token if the token is older than 80 percent of its time to live or if the token is older than 24 hours. Defaults to 1 hour and must be at least 10 minutes.
path	string	path is the path relative to the mount point of the file to project the token into.

3.1.311. .template.spec.volumes[].quobyte

Description

Represents a Quobyte mount that lasts the lifetime of a pod. Quobyte volumes do not support ownership management or SELinux relabeling.

Type

object

Required

- **registry**
- **volume**

Property	Type	Description
group	string	group to map volume access to Default is no group

Property	Type	Description
readOnly	boolean	readOnly here will force the Quobyte volume to be mounted with read-only permissions. Defaults to false.
registry	string	registry represents a single or multiple Quobyte Registry services specified as a string as host:port pair (multiple entries are separated with commas) which acts as the central registry for volumes
tenant	string	tenant owning the given Quobyte volume in the Backend Used with dynamically provisioned Quobyte volumes, value is set by the plugin
user	string	user to map volume access to Defaults to serviceaccount user
volume	string	volume is a string that references an already created Quobyte volume by name.

3.1.312. .template.spec.volumes[].rbd

Description

Represents a Rados Block Device mount that lasts the lifetime of a pod. RBD volumes support ownership management and SELinux relabeling.

Type

object

Required

- **monitors**
- **image**

Property	Type	Description
----------	------	-------------

Property	Type	Description
fsType	string	fsType is the filesystem type of the volume that you want to mount. Tip: Ensure that the filesystem type is supported by the host operating system. Examples: "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified. More info: https://kubernetes.io/docs/concepts/storage/volumes#rbd
image	string	image is the rados image name. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
keyring	string	keyring is the path to key ring for RBDUser. Default is /etc/ceph/keyring. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
monitors	array (string)	monitors is a collection of Ceph monitors. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
pool	string	pool is the rados pool name. Default is rbd. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
readOnly	boolean	readOnly here will force the ReadOnly setting in VolumeMounts. Defaults to false. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
user	string	user is the rados user name. Default is admin. More info: https://examples.k8s.io/volumes/rbd/README.md#how-to-use-it

3.1.313. .template.spec.volumes[].rbd.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

3.1.314. .template.spec.volumes[].scaleIO

Description

ScaleIOVolumeSource represents a persistent ScaleIO volume

Type

object

Required

- **gateway**
- **system**
- **secretRef**

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Default is "xfs".
gateway	string	gateway is the host address of the ScaleIO API Gateway.
protectionDomain	string	protectionDomain is the name of the ScaleIO Protection Domain for the configured storage.

Property	Type	Description
readOnly	boolean	readOnly Defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
sslEnabled	boolean	sslEnabled Flag enable/disable SSL communication with Gateway, default false
storageMode	string	storageMode indicates whether the storage for a volume should be ThickProvisioned or ThinProvisioned. Default is ThinProvisioned.
storagePool	string	storagePool is the ScaleIO Storage Pool associated with the protection domain.
system	string	system is the name of the storage system as configured in ScaleIO.
volumeName	string	volumeName is the name of a volume already created in the ScaleIO system that is associated with this volume source.

3.1.315. .template.spec.volumes[].scaleIO.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
----------	------	-------------

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

3.1.316. .template.spec.volumes[].secret

Description

Adapts a Secret into a volume.

The contents of the target Secret's Data field will be presented in a volume as files using the keys in the Data field as the file names. Secret volumes support ownership management and SELinux relabeling.

Type

object

Property	Type	Description
defaultMode	integer	defaultMode is Optional: mode bits used to set permissions on created files by default. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. Defaults to 0644. Directories within the path are not affected by this setting. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.

Property	Type	Description
items	array	items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.
items[]	object	Maps a string key to a path within a volume.
optional	boolean	optional field specify whether the Secret or its keys must be defined
secretName	string	secretName is the name of the secret in the pod's namespace to use. More info: https://kubernetes.io/docs/concepts/storage/volumes#secret

3.1.317. .template.spec.volumes[].secret.items

Description

items If unspecified, each key-value pair in the Data field of the referenced Secret will be projected into the volume as a file whose name is the key and content is the value. If specified, the listed keys will be projected into the specified paths, and unlisted keys will not be present. If a key is specified which is not present in the Secret, the volume setup will error unless it is marked optional. Paths must be relative and may not contain the '..' path or start with '..'.

Type

array

3.1.318. .template.spec.volumes[].secret.items[]

Description

Maps a string key to a path within a volume.

Type

object

Required

- **key**
- **path**

Property	Type	Description
key	string	key is the key to project.
mode	integer	mode is Optional: mode bits used to set permissions on this file. Must be an octal value between 0000 and 0777 or a decimal value between 0 and 511. YAML accepts both octal and decimal values, JSON requires decimal values for mode bits. If not specified, the volume defaultMode will be used. This might be in conflict with other options that affect the file mode, like fsGroup, and the result can be other mode bits set.
path	string	path is the relative path of the file to map the key to. May not be an absolute path. May not contain the path element '..'. May not start with the string '..'.

3.1.319. .template.spec.volumes[].storageos

Description

Represents a StorageOS persistent volume resource.

Type

object

Property	Type	Description
fsType	string	fsType is the filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
readOnly	boolean	readOnly defaults to false (read/write). ReadOnly here will force the ReadOnly setting in VolumeMounts.

Property	Type	Description
secretRef	object	LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.
volumeName	string	volumeName is the human-readable name of the StorageOS volume. Volume names are only unique within a namespace.
volumeNamespace	string	volumeNamespace specifies the scope of the volume within StorageOS. If no namespace is specified then the Pod's namespace will be used. This allows the Kubernetes name scoping to be mirrored within StorageOS for tighter integration. Set VolumeName to any name to override the default behaviour. Set to "default" if you are not using namespaces within StorageOS. Namespaces that do not pre-exist within StorageOS will be created.

3.1.320. .template.spec.volumes[].storageos.secretRef

Description

LocalObjectReference contains enough information to let you locate the referenced object inside the same namespace.

Type

object

Property	Type	Description
name	string	Name of the referent. This field is effectively required, but due to backwards compatibility is allowed to be empty. Instances of this type with an empty value here are almost certainly wrong. More info: https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#names

3.1.321. .template.spec.volumes[].vsphereVolume

Description

Represents a vSphere volume resource.

Type

object

Required

- **volumePath**

Property	Type	Description
fsType	string	fsType is filesystem type to mount. Must be a filesystem type supported by the host operating system. Ex. "ext4", "xfs", "ntfs". Implicitly inferred to be "ext4" if unspecified.
storagePolicyID	string	storagePolicyID is the storage Policy Based Management (SPBM) profile ID associated with the StoragePolicyName.
storagePolicyName	string	storagePolicyName is the storage Policy Based Management (SPBM) profile name.
volumePath	string	volumePath is the path that identifies vSphere volume vmdk

3.2. API ENDPOINTS

The following API endpoints are available:

- **/api/v1/podtemplates**
 - **GET**: list or watch objects of kind PodTemplate
- **/api/v1/watch/podtemplates**
 - **GET**: watch individual changes to a list of PodTemplate. deprecated: use the 'watch' parameter with a list operation instead.
- **/api/v1/namespaces/{namespace}/podtemplates**
 - **DELETE**: delete collection of PodTemplate
 - **GET**: list or watch objects of kind PodTemplate
 - **POST**: create a PodTemplate
- **/api/v1/watch/namespaces/{namespace}/podtemplates**

- **GET**: watch individual changes to a list of PodTemplate. deprecated: use the 'watch' parameter with a list operation instead.
- **/api/v1/namespaces/{namespace}/podtemplates/{name}**
 - **DELETE**: delete a PodTemplate
 - **GET**: read the specified PodTemplate
 - **PATCH**: partially update the specified PodTemplate
 - **PUT**: replace the specified PodTemplate
- **/api/v1/watch/namespaces/{namespace}/podtemplates/{name}**
 - **GET**: watch changes to an object of kind PodTemplate. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

3.2.1. /api/v1/podtemplates

HTTP method

GET

Description

list or watch objects of kind PodTemplate

Table 3.1. HTTP responses

HTTP code	Reponse body
200 - OK	PodTemplateList schema
401 - Unauthorized	Empty

3.2.2. /api/v1/watch/podtemplates

HTTP method

GET

Description

watch individual changes to a list of PodTemplate. deprecated: use the 'watch' parameter with a list operation instead.

Table 3.2. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

3.2.3. /api/v1/namespaces/{namespace}/podtemplates

HTTP method

DELETE

Description

delete collection of PodTemplate

Table 3.3. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 3.4. HTTP responses

HTTP code	Reponse body
200 - OK	Status schema
401 - Unauthorized	Empty

HTTP method

GET

Description

list or watch objects of kind PodTemplate

Table 3.5. HTTP responses

HTTP code	Reponse body
200 - OK	PodTemplateList schema
401 - Unauthorized	Empty

HTTP method

POST

Description

create a PodTemplate

Table 3.6. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 3.7. Body parameters

Parameter	Type	Description
body	PodTemplate schema	

Table 3.8. HTTP responses

HTTP code	Response body
200 - OK	PodTemplate schema
201 - Created	PodTemplate schema
202 - Accepted	PodTemplate schema
401 - Unauthorized	Empty

3.2.4. /api/v1/watch/namespaces/{namespace}/podtemplates

HTTP method

GET**Description**

watch individual changes to a list of PodTemplate. deprecated: use the 'watch' parameter with a list operation instead.

Table 3.9. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

3.2.5. /api/v1/namespaces/{namespace}/podtemplates/{name}**Table 3.10. Global path parameters**

Parameter	Type	Description
name	string	name of the PodTemplate

HTTP method**DELETE****Description**

delete a PodTemplate

Table 3.11. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 3.12. HTTP responses

HTTP code	Reponse body
200 - OK	PodTemplate schema
202 - Accepted	PodTemplate schema
401 - Unauthorized	Empty

HTTP method

GET**Description**

read the specified PodTemplate

Table 3.13. HTTP responses

HTTP code	Response body
200 - OK	PodTemplate schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update the specified PodTemplate

Table 3.14. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 3.15. HTTP responses

HTTP code	Response body
200 - OK	PodTemplate schema
201 - Created	PodTemplate schema
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace the specified PodTemplate

Table 3.16. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 3.17. Body parameters

Parameter	Type	Description
body	PodTemplate schema	

Table 3.18. HTTP responses

HTTP code	Reponse body
200 - OK	PodTemplate schema
201 - Created	PodTemplate schema
401 - Unauthorized	Empty

3.2.6. /api/v1/watch/namespaces/{namespace}/podtemplates/{name}

Table 3.19. Global path parameters

Parameter	Type	Description
name	string	name of the PodTemplate

HTTP method

GET

Description

watch changes to an object of kind PodTemplate. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 3.20. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

CHAPTER 4. TEMPLATE [TEMPLATE.OPENSIFT.IO/V1]

Description

Template contains the inputs needed to produce a Config.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **objects**

4.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
labels	object (string)	labels is a optional set of labels that are applied to every object during the Template to Config transformation.

Property	Type	Description
message	string	message is an optional instructional message that will be displayed when this template is instantiated. This field should inform the user how to utilize the newly created resources. Parameter substitution will be performed on the message before being displayed so that generated credentials and other parameters can be included in the output.
metadata	ObjectMeta_v2	metadata is the standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata
objects	array (RawExtension)	objects is an array of resources to include in this template. If a namespace value is hardcoded in the object, it will be removed during template instantiation, however if the namespace value is, or contains, a <code>\${PARAMETER_REFERENCE}</code> , the resolved value after parameter substitution will be respected and the object will be created in that namespace.
parameters	array	parameters is an optional array of Parameters used during the Template to Config transformation.
parameters[]	object	Parameter defines a name/value variable that is to be processed during the Template to Config transformation.

4.1.1. .parameters

Description

parameters is an optional array of Parameters used during the Template to Config transformation.

Type

array

4.1.2. .parameters[]

Description

Parameter defines a name/value variable that is to be processed during the Template to Config transformation.

Type

object

Required

- **name**

Property	Type	Description
description	string	Description of a parameter. Optional.
displayName	string	Optional: The name that will show in UI instead of parameter 'Name'
from	string	From is an input value for the generator. Optional.
generate	string	<p>generate specifies the generator to be used to generate random string from an input value specified by From field. The result string is stored into Value field. If empty, no generator is being used, leaving the result Value untouched. Optional.</p> <p>The only supported generator is "expression", which accepts a "from" value in the form of a simple regular expression containing the range expression "[a-zA-Z0-9]", and the length expression "a{length}".</p> <p>Examples:</p> <pre>from value ----- ----- "test[0-9]{1}x" "test7x" "[0-1]{8}" "01001100" "0x[A-F0-9]{4}" "0xB3AF" "[a- zA-Z0-9]{8}" "hW4yQU5i"</pre>
name	string	Name must be set and it can be referenced in Template Items using \${PARAMETER_NAME}. Required.

Property	Type	Description
required	boolean	Optional: Indicates the parameter must have a value. Defaults to false.
value	string	Value holds the Parameter data. If specified, the generator will be ignored. The value replaces all occurrences of the Parameter <code>\${Name}</code> expression during the Template to Config transformation. Optional.

4.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/template.openshift.io/v1/templates**
 - **GET**: list or watch objects of kind Template
- **/apis/template.openshift.io/v1/watch/templates**
 - **GET**: watch individual changes to a list of Template. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/template.openshift.io/v1/namespaces/{namespace}/templates**
 - **DELETE**: delete collection of Template
 - **GET**: list or watch objects of kind Template
 - **POST**: create a Template
- **/apis/template.openshift.io/v1/watch/namespaces/{namespace}/templates**
 - **GET**: watch individual changes to a list of Template. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/template.openshift.io/v1/namespaces/{namespace}/templates/{name}**
 - **DELETE**: delete a Template
 - **GET**: read the specified Template
 - **PATCH**: partially update the specified Template
 - **PUT**: replace the specified Template
- **/apis/template.openshift.io/v1/namespaces/{namespace}/processedtemplates**
 - **POST**: create a Template
- **/apis/template.openshift.io/v1/watch/namespaces/{namespace}/templates/{name}**

- **GET**: watch changes to an object of kind Template. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

4.2.1. /apis/template.openshift.io/v1/templates

HTTP method

GET

Description

list or watch objects of kind Template

Table 4.1. HTTP responses

HTTP code	Reponse body
200 - OK	TemplateList schema
401 - Unauthorized	Empty

4.2.2. /apis/template.openshift.io/v1/watch/templates

HTTP method

GET

Description

watch individual changes to a list of Template. deprecated: use the 'watch' parameter with a list operation instead.

Table 4.2. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

4.2.3. /apis/template.openshift.io/v1/namespaces/{namespace}/templates

HTTP method

DELETE

Description

delete collection of Template

Table 4.3. Query parameters

Parameter	Type	Description
-----------	------	-------------

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 4.4. HTTP responses

HTTP code	Response body
200 - OK	Status_v9 schema
401 - Unauthorized	Empty

HTTP method**GET****Description**

list or watch objects of kind Template

Table 4.5. HTTP responses

HTTP code	Response body
200 - OK	TemplateList schema
401 - Unauthorized	Empty

HTTP method**POST****Description**

create a Template

Table 4.6. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 4.7. Body parameters

Parameter	Type	Description
body	Template schema	

Table 4.8. HTTP responses

HTTP code	Reponse body
200 - OK	Template schema
201 - Created	Template schema
202 - Accepted	Template schema
401 - Unauthorized	Empty

4.2.4. /apis/template.openshift.io/v1/watch/namespaces/{namespace}/templates

HTTP method

GET

Description

watch individual changes to a list of Template. deprecated: use the 'watch' parameter with a list operation instead.

Table 4.9. HTTP responses

HTTP code	Response body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

4.2.5. /apis/template.openshift.io/v1/namespaces/{namespace}/templates/{name}

Table 4.10. Global path parameters

Parameter	Type	Description
name	string	name of the Template

HTTP method

DELETE

Description

delete a Template

Table 4.11. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 4.12. HTTP responses

HTTP code	Response body
200 - OK	Template schema
202 - Accepted	Template schema
401 - Unauthorized	Empty

HTTP method

GET

Description

read the specified Template

Table 4.13. HTTP responses

HTTP code	Reponse body
200 - OK	Template schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update the specified Template

Table 4.14. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 4.15. HTTP responses

HTTP code	Reponse body
200 - OK	Template schema
201 - Created	Template schema

HTTP code	Response body
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace the specified Template

Table 4.16. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 4.17. Body parameters

Parameter	Type	Description
body	Template schema	

Table 4.18. HTTP responses

HTTP code	Reponse body
200 - OK	Template schema
201 - Created	Template schema
401 - Unauthorized	Empty

4.2.6. /apis/template.openshift.io/v1/namespaces/{namespace}/processedtemplates

Table 4.19. Global query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

HTTP method

POST

Description

create a Template

Table 4.20. Body parameters

Parameter	Type	Description
body	Template schema	

Table 4.21. HTTP responses

HTTP code	Reponse body
200 - OK	Template schema
201 - Created	Template schema
202 - Accepted	Template schema
401 - Unauthorized	Empty

4.2.7. /apis/template.openshift.io/v1/watch/namespaces/{namespace}/templates/{n

Table 4.22. Global path parameters

Parameter	Type	Description
name	string	name of the Template

HTTP method

GET

Description

watch changes to an object of kind Template. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 4.23. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

CHAPTER 5. TEMPLATEINSTANCE [TEMPLATE.OPENSIFT.IO/V1]

Description

TemplateInstance requests and records the instantiation of a Template. TemplateInstance is part of an experimental API.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **spec**

5.1. SPECIFICATION

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
metadata	ObjectMeta_v2	metadata is the standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

Property	Type	Description
spec	object	TemplateInstanceSpec describes the desired state of a TemplateInstance.
status	object	TemplateInstanceStatus describes the current state of a TemplateInstance.

5.1.1. .spec

Description

TemplateInstanceSpec describes the desired state of a TemplateInstance.

Type

object

Required

- **template**

Property	Type	Description
requester	object	TemplateInstanceRequester holds the identity of an agent requesting a template instantiation.
secret	LocalObjectReference_v2	secret is a reference to a Secret object containing the necessary template parameters.
template	object	<p>Template contains the inputs needed to produce a Config.</p> <p>Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).</p>

5.1.2. .spec.requester

Description

TemplateInstanceRequester holds the identity of an agent requesting a template instantiation.

Type

object

Property	Type	Description
extra	object	extra holds additional information provided by the authenticator.
extra{}	array (string)	
groups	array (string)	groups represent the groups this user is a part of.
uid	string	uid is a unique value that identifies this user across time; if this user is deleted and another user by the same name is added, they will have different UIDs.
username	string	username uniquely identifies this user among all active users.

5.1.3. .spec.requester.extra

Description

extra holds additional information provided by the authenticator.

Type

object

5.1.4. .spec.template

Description

Template contains the inputs needed to produce a Config.

Compatibility level 1: Stable within a major release for a minimum of 12 months or 3 minor releases (whichever is longer).

Type

object

Required

- **objects**

Property	Type	Description
----------	------	-------------

Property	Type	Description
apiVersion	string	APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources
kind	string	Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds
labels	object (string)	labels is a optional set of labels that are applied to every object during the Template to Config transformation.
message	string	message is an optional instructional message that will be displayed when this template is instantiated. This field should inform the user how to utilize the newly created resources. Parameter substitution will be performed on the message before being displayed so that generated credentials and other parameters can be included in the output.
metadata	ObjectMeta_v2	metadata is the standard object's metadata. More info: https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#metadata

Property	Type	Description
objects	array (RawExtension)	objects is an array of resources to include in this template. If a namespace value is hardcoded in the object, it will be removed during template instantiation, however if the namespace value is, or contains, a <code>\${PARAMETER_REFERENCE}</code> , the resolved value after parameter substitution will be respected and the object will be created in that namespace.
parameters	array	parameters is an optional array of Parameters used during the Template to Config transformation.
parameters[]	object	Parameter defines a name/value variable that is to be processed during the Template to Config transformation.

5.1.5. .spec.template.parameters

Description

parameters is an optional array of Parameters used during the Template to Config transformation.

Type

array

5.1.6. .spec.template.parameters[]

Description

Parameter defines a name/value variable that is to be processed during the Template to Config transformation.

Type

object

Required

- **name**

Property	Type	Description
description	string	Description of a parameter. Optional.

Property	Type	Description
displayName	string	Optional: The name that will show in UI instead of parameter 'Name'
from	string	From is an input value for the generator. Optional.
generate	string	<p>generate specifies the generator to be used to generate random string from an input value specified by From field. The result string is stored into Value field. If empty, no generator is being used, leaving the result Value untouched. Optional.</p> <p>The only supported generator is "expression", which accepts a "from" value in the form of a simple regular expression containing the range expression "[a-zA-Z0-9]", and the length expression "a{length}".</p> <p>Examples:</p> <pre> from value ----- ----- "test[0-9]{1}x" "test7x" "[0-1]{8}" "01001100" "0x[A-F0-9]{4}" "0xB3AF" "[a- zA-Z0-9]{8}" "hW4yQU5i" </pre>
name	string	Name must be set and it can be referenced in Template Items using \${PARAMETER_NAME}. Required.
required	boolean	Optional: Indicates the parameter must have a value. Defaults to false.
value	string	Value holds the Parameter data. If specified, the generator will be ignored. The value replaces all occurrences of the Parameter \${Name} expression during the Template to Config transformation. Optional.

5.1.7. .status

Description

TemplateInstanceStatus describes the current state of a TemplateInstance.

Type

object

Property	Type	Description
conditions	array	conditions represent the latest available observations of a TemplateInstance's current state.
conditions[]	object	TemplateInstanceCondition contains condition information for a TemplateInstance.
objects	array	Objects references the objects created by the TemplateInstance.
objects[]	object	TemplateInstanceObject references an object created by a TemplateInstance.

5.1.8. .status.conditions

Description

conditions represent the latest available observations of a TemplateInstance's current state.

Type

array

5.1.9. .status.conditions[]

Description

TemplateInstanceCondition contains condition information for a TemplateInstance.

Type

object

Required

- **type**
- **status**
- **lastTransitionTime**
- **reason**
- **message**

Property	Type	Description
lastTransitionTime	Time	LastTransitionTime is the last time a condition status transitioned from one state to another.
message	string	Message is a human readable description of the details of the last transition, complementing reason.
reason	string	Reason is a brief machine readable explanation for the condition's last transition.
status	string	Status of the condition, one of True, False or Unknown.
type	string	Type of the condition, currently Ready or InstantiateFailure.

5.1.10. .status.objects

Description

Objects references the objects created by the TemplateInstance.

Type

array

5.1.11. .status.objects[]

Description

TemplateInstanceObject references an object created by a TemplateInstance.

Type

object

Property	Type	Description
ref	ObjectReference	ref is a reference to the created object. When used under .spec, only name and namespace are used; these can contain references to parameters which will be substituted following the usual rules.

5.2. API ENDPOINTS

The following API endpoints are available:

- **/apis/template.openshift.io/v1/templateinstances**
 - **GET**: list or watch objects of kind TemplateInstance
- **/apis/template.openshift.io/v1/watch/templateinstances**
 - **GET**: watch individual changes to a list of TemplateInstance. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/template.openshift.io/v1/namespaces/{namespace}/templateinstances**
 - **DELETE**: delete collection of TemplateInstance
 - **GET**: list or watch objects of kind TemplateInstance
 - **POST**: create a TemplateInstance
- **/apis/template.openshift.io/v1/watch/namespaces/{namespace}/templateinstances**
 - **GET**: watch individual changes to a list of TemplateInstance. deprecated: use the 'watch' parameter with a list operation instead.
- **/apis/template.openshift.io/v1/namespaces/{namespace}/templateinstances/{name}**
 - **DELETE**: delete a TemplateInstance
 - **GET**: read the specified TemplateInstance
 - **PATCH**: partially update the specified TemplateInstance
 - **PUT**: replace the specified TemplateInstance
- **/apis/template.openshift.io/v1/watch/namespaces/{namespace}/templateinstances/{name}**
 - **GET**: watch changes to an object of kind TemplateInstance. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.
- **/apis/template.openshift.io/v1/namespaces/{namespace}/templateinstances/{name}/status**
 - **GET**: read status of the specified TemplateInstance
 - **PATCH**: partially update status of the specified TemplateInstance
 - **PUT**: replace status of the specified TemplateInstance

5.2.1. /apis/template.openshift.io/v1/templateinstances

HTTP method

GET

Description

list or watch objects of kind TemplateInstance

Table 5.1. HTTP responses

HTTP code	Reponse body
200 - OK	TemplateInstanceList schema
401 - Unauthorized	Empty

5.2.2. /apis/template.openshift.io/v1/watch/templateinstances

HTTP method

GET

Description

watch individual changes to a list of TemplateInstance. deprecated: use the 'watch' parameter with a list operation instead.

Table 5.2. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

5.2.3. /apis/template.openshift.io/v1/namespaces/{namespace}/templateinstances

HTTP method

DELETE

Description

delete collection of TemplateInstance

Table 5.3. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 5.4. HTTP responses

HTTP code	Reponse body
200 - OK	Status_v9 schema
401 - Unauthorized	Empty

HTTP method**GET****Description**

list or watch objects of kind TemplateInstance

Table 5.5. HTTP responses

HTTP code	Response body
200 - OK	TemplateInstanceList schema
401 - Unauthorized	Empty

HTTP method**POST****Description**

create a TemplateInstance

Table 5.6. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 5.7. Body parameters

Parameter	Type	Description
body	TemplateInstance schema	

Table 5.8. HTTP responses

HTTP code	Reponse body
200 - OK	TemplateInstance schema
201 - Created	TemplateInstance schema
202 - Accepted	TemplateInstance schema
401 - Unauthorized	Empty

5.2.4. /apis/template.openshift.io/v1/watch/namespaces/{namespace}/templateinstanc

HTTP method

GET

Description

watch individual changes to a list of TemplateInstance. deprecated: use the 'watch' parameter with a list operation instead.

Table 5.9. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

5.2.5. /apis/template.openshift.io/v1/namespaces/{namespace}/templateinstances/{

Table 5.10. Global path parameters

Parameter	Type	Description
name	string	name of the TemplateInstance

HTTP method

DELETE

Description

delete a TemplateInstance

Table 5.11. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Table 5.12. HTTP responses

HTTP code	Reponse body
200 - OK	Status_v9 schema
202 - Accepted	Status_v9 schema
401 - Unauthorized	Empty

HTTP method**GET****Description**

read the specified TemplateInstance

Table 5.13. HTTP responses

HTTP code	Reponse body
200 - OK	TemplateInstance schema
401 - Unauthorized	Empty

HTTP method**PATCH****Description**

partially update the specified TemplateInstance

Table 5.14. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 5.15. HTTP responses

HTTP code	Response body
200 - OK	TemplateInstance schema
201 - Created	TemplateInstance schema
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace the specified TemplateInstance

Table 5.16. Query parameters

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: <ul style="list-style-type: none"> - All: all dry run stages will be processed

Parameter	Type	Description
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: <ul style="list-style-type: none"> - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 5.17. Body parameters

Parameter	Type	Description
body	TemplateInstance schema	

Table 5.18. HTTP responses

HTTP code	Response body
200 - OK	TemplateInstance schema
201 - Created	TemplateInstance schema
401 - Unauthorized	Empty

5.2.6. /apis/template.openshift.io/v1/watch/namespaces/{namespace}/templateinstances

Table 5.19. Global path parameters

Parameter	Type	Description
name	string	name of the TemplateInstance

HTTP method

GET

Description

watch changes to an object of kind TemplateInstance. deprecated: use the 'watch' parameter with a list operation instead, filtered to a single item with the 'fieldSelector' parameter.

Table 5.20. HTTP responses

HTTP code	Reponse body
200 - OK	WatchEvent schema
401 - Unauthorized	Empty

5.2.7. /apis/template.openshift.io/v1/namespaces/{namespace}/templateinstances/{

Table 5.21. Global path parameters

Parameter	Type	Description
name	string	name of the TemplateInstance

HTTP method

GET

Description

read status of the specified TemplateInstance

Table 5.22. HTTP responses

HTTP code	Reponse body
200 - OK	TemplateInstance schema
401 - Unauthorized	Empty

HTTP method

PATCH

Description

partially update status of the specified TemplateInstance

Table 5.23. Query parameters

Parameter	Type	Description
-----------	------	-------------

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 5.24. HTTP responses

HTTP code	Response body
200 - OK	TemplateInstance schema
201 - Created	TemplateInstance schema
401 - Unauthorized	Empty

HTTP method**PUT****Description**

replace status of the specified TemplateInstance

Table 5.25. Query parameters

Parameter	Type	Description
-----------	------	-------------

Parameter	Type	Description
dryRun	string	When present, indicates that modifications should not be persisted. An invalid or unrecognized dryRun directive will result in an error response and no further processing of the request. Valid values are: - All: all dry run stages will be processed
fieldValidation	string	fieldValidation instructs the server on how to handle objects in the request (POST/PUT/PATCH) containing unknown or duplicate fields. Valid values are: - Ignore: This will ignore any unknown fields that are silently dropped from the object, and will ignore all but the last duplicate field that the decoder encounters. This is the default behavior prior to v1.23. - Warn: This will send a warning via the standard warning response header for each unknown field that is dropped from the object, and for each duplicate field that is encountered. The request will still succeed if there are no other errors, and will only persist the last of any duplicate fields. This is the default in v1.23+ - Strict: This will fail the request with a BadRequest error if any unknown fields would be dropped from the object, or if any duplicate fields are present. The error returned from the server will contain all unknown and duplicate fields encountered.

Table 5.26. Body parameters

Parameter	Type	Description
body	TemplateInstance schema	

Table 5.27. HTTP responses

HTTP code	Reponse body
200 - OK	TemplateInstance schema
201 - Created	TemplateInstance schema
401 - Unauthorized	Empty