

containers:

- **args**

Arguments to the entrypoint. The docker image's CMD is used if this is not provided.

- **command** <string>

Entrypoint array. Not executed within a shell. The docker image's ENTRYPOINT is used if this is not provided.

- **env** <[]Object>

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

- **name** <string> -required-

Name of the environment variable. Must be a C_IDENTIFIER.

- **value** <string>

Variable references \$(VAR_NAME) are expanded using the previously defined environment variables in the container and any service environment variables.

- **valueFrom** <Object>

Source for the environment variable's value. Cannot be used if value is not empty.

- **configMapKeyRef** <Object>

Selects a key of a ConfigMap.

- **key** <string> -required-

The key to select.

- **name** <string>

Name of the referent. More info:

- **optional** <boolean>

Specify whether the ConfigMap or its key must be defined

- **fieldRef** <Object>

Selects a field of the pod: supports metadata.name, metadata.namespace, `metadata.labels['<KEY>']`, `metadata.annotations['<KEY>']`, spec.nodeName, spec.serviceAccountName, status.hostIP, status.podIP, status.podIPs.

- **apiVersion** <string>

Version of the schema the FieldPath is written in terms of, defaults to "v1".

- **fieldPath** <string> -required-

Path of the field to select in the specified API version.

- **resourceFieldRef** <Object>

Selects a resource of the container: only resources limits and requests (limits.cpu, limits.memory, limits.ephemeral-storage, requests.cpu, requests.memory and requests.ephemeral-storage) are currently supported.

- **containerName** <string>

Container name: required for volumes, optional for env vars

- **divisor** <string>

Specifies the output format of the exposed resources, defaults to "1"

- **resource** <string> -required-

Required: resource to select

- **secretKeyRef** <Object>

Selects a key of a secret in the pod's namespace

- **key** <string> -required-
The key of the secret to select from. Must be a valid secret key.
- **name** <string>
Name of the referent. More info:
- **optional** <boolean>
Specify whether the Secret or its key must be defined

- **envFrom** <Object>

List of sources to populate environment variables in the container.

EnvFromSource represents the source of a set of ConfigMaps

- **configMapRef** <Object>
The ConfigMap to select from
- **prefix** <string>
An optional identifier to prepend to each key in the ConfigMap. Must be a C_IDENTIFIER.
- **secretRef** <Object>
The Secret to select from

- **name** <string> -required-

- **image** <string>

Docker image name.

- **imagePullPolicy** <string>

Image pull policy. One of Always, Never, IfNotPresent. Defaults to Always if :latest tag is specified, or IfNotPresent otherwise. Cannot be updated.

- **lifecycle** <Object>

Actions that the management system should take in response to container lifecycle events. Cannot be updated.

- **livenessProbe** <Object>

Periodic probe of container liveness. Container will be restarted if the probe fails. Cannot be updated.

- **exec** <OBJECT>
One and only one of the following should be specified. Exec specifies the action to take.
- **failureThreshold** <integer>
Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.
- **httpGet** <Object>
HTTPGet specifies the http request to perform.
- **initialDelaySeconds** <integer>
Number of seconds after the container has started before liveness probes are initiated.
- **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.
- **timeoutSeconds** <integer>
Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1.

- **ports** <Object>

List of ports to expose from the container. Exposing a port here gives the system additional information about the network connections a container uses, but is primarily informational.

- **containerPort** <integer> -required-
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".

- **readinessProbe** <Object>

Periodic probe of container service readiness. Container will be removed from service endpoints if the probe fails. Cannot be updated.

- **exec** <Object>

One and only one of the following should be specified. Exec specifies the action to take.

- **failureThreshold** <integer>

Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.

- **httpGet** <Object>

HTTPGet specifies the http request to perform.

- **initialDelaySeconds** <integer>

Number of seconds after the container has started before liveness probes are initiated. More info:

- **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>

TCP socket specifies an action involving a TCP port. TCP hooks not yet supported

- **timeoutSeconds** <integer>

Number of seconds after which the probe times out. Defaults to 1 second. Minimum value is 1.

- **resources** <Object>

Compute Resources required by this container. Cannot be updated.

- **limits** <map[string]string>

Limits describes the maximum amount of compute resources allowed.

- **requests** <map[string]string>

Requests describes the minimum amount of compute resources required.

- **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.

- **terminationMessagePolicy** <string>

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.

- **securityContext** <Object>

Security options the pod should run with.

- **allowPrivilegeEscalation** <boolean>

AllowPrivilegeEscalation controls whether a process can gain more privileges than its parent process.

- **capabilities** <Object>

The capabilities to add/drop when running containers. Defaults to the default set of capabilities granted by the container runtime.

- **privileged** <boolean>

Run container in privileged mode. Processes in privileged containers are essentially equivalent to root on the host. Defaults to false.

- **procMount** <string>

procMount denotes the type of proc mount to use for the containers.

- **readOnlyRootFilesystem** <boolean>

Whether this container has a read-only root filesystem. Default is false.

- **runAsGroup** <integer>

The GID to run the entrypoint of the container process. Uses runtime default if unset. .

- **runAsNonRoot** <boolean>

Indicates that the container must run as a non-root user.

- **runAsUser** <integer>

The UID to run the entrypoint of the container process. Defaults to user specified in image metadata if unspecified.

- **seLinuxOptions** <Object>

The SELinux context to be applied to the container.

- **seccompProfile** <Object>

The seccomp options to use by this container. If seccomp options are provided at both the pod & container level, the container options override the pod options.

- **windowsOptions** <Object>

The Windows specific settings applied to all containers.

- **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.

- **tty** <boolean>

Whether this container should allocate a TTY for itself, also requires 'stdin' to be true. Default is false.

- **startupProbe** <Object>

StartupProbe indicates that the Pod has successfully initialized. If specified, no other probes are executed until this completes successfully.

- **exec** <Object>

One and only one of the following should be specified. Exec specifies the action to take.

- **failureThreshold** <integer>

Minimum consecutive failures for the probe to be considered failed after having succeeded. Defaults to 3. Minimum value is 1.

- **httpGet** <Object>

HTTPGet specifies the http request to perform.

- **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>

TCP socket specifies an action involving a TCP port. TCP hooks not yet supported

- **timeoutSeconds** <integer>

Number of seconds after which the probe times out. Defaults to 1 second.

Minimum value is 1.

- **volumeMounts** <[]Object>

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

- **mountPath** <string> -required-

Path within the container at which the volume should be mounted. Must not contain ':

- **mountPropagation** <string>

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.

- **name** <string> -required-

This must match the Name of a Volume.

- **readOnly** <boolean>

Mounted read-only if true, read-write otherwise (false or unspecified).

Defaults to false.

- **subPath** <string>

Path within the volume from which the container's volume should be mounted.

Defaults to "" (volume's root).

- **subPathExpr** <string>

Expanded path within the volume from which the container's volume should be mounted.

- **volumeDevices** <Object>

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