CLAIMS

1. A computer-implemented method executed by a server in a cluster for checking an integrity of an object to be deployed to the cluster, comprising:

detecting a resource creation request from a client;

responsive to the resource creation request being an initial resource creation request for the object, verifying the integrity of the object based on properties in the resource creation request to create a release secret in the cluster for a positive integrity verification result for the object, the release secret representing a specific deployment configuration of the object on the cluster;

responsive to the resource creation request being other than the initial resource request, checking if the resource creation request corresponds to the specific deployment configuration of the object by checking against the release secret in the cluster; and

responsive to the resource creation request corresponding to a deployment of the object and the release secret being present in the cluster, creating a resource requested by the resource creation request in the cluster.

2. The computer-implemented method of claim 1, wherein verifying the integrity of the object comprises:

checking a name of the resource requested by the resource creation request;

checking a signature on an object content retrieved from a repository; and

matching between a template definition in the signature on the object and a definition in the resource creation request; and matching between a signed deployment configuration in a database and a deployment configuration in the resource creation request, responsive to the name of the resource and the signature being found.

3. The computer-implemented method of claim 1, wherein checking if the resource creation request corresponds to the specific deployment configuration of the object comprises:

creating an expected resource definition from a signed template and the specific deployment configuration; and

comparing the expected resource definition with a resource definition in the resource creation request.

4. The computer-implemented method of claim 1, further comprising:

checking if the resource creation request corresponds to the specific deployment configuration of the object by checking if an identifier of an owner of object content of the object is found in the cluster;

responsive to the identifier of the owner being found in the cluster, verifying an integrity of the object and creating the resource with an annotation in the cluster, the annotation indicating the positive integrity verification result; and

blocking any incoming update requests responsive to the annotation lacking being found in the resource.

5. The computer-implemented method of claim 1, wherein verifying the integrity of the object is performed solely at the cluster.

6. The computer-implemented method of claim 1, wherein verifying the integrity of the object is performed using a mutating admission webhook in the cluster.

7. The computer-implemented method of claim 6, wherein the mutating admission webhook intercepts the resource creation request to impose an integrity check on the resource creation request.

8. The computer-implemented method of claim 1, wherein verifying the integrity of the object using the secret release is performed for both creation of the object and updating of the object.

9. The computer-implemented method of claim 1, wherein verifying the integrity of the object includes adding an annotation to indicating the positive integrity verification result for a package.

10. The computer-implemented method of claim 1, wherein verifying the integrity of the object uses a signature of a resource signature.

11. The computer-implemented method of claim 10, wherein the signature is detected as an artifact associated with the specific deployment configuration.

12. The computer-implemented method of claim 1, wherein the release secret stores a release embodying a running instance of the object in the cluster.

13. The computer-implemented method of claim 1, wherein verifying the integrity of the object comprises:

checking whether an authorized owner signed a chart corresponding to the object and found in a repository; and

checking whether a content of the chart was untampered with since the content of the chart was bundled.

14. The computer-implemented method of claim 13, wherein checking whether the content of the chart was tampered with comprises comparing a content from a release source in the resource creation request with expected content from the chart found in the repository.

15. The computer-implemented method of claim 1, further comprising handling variations in object packaging and deployments by ensuring the variations are associated with, and verified against, a signature from a repository or in the cluster.

16. The computer-implemented method of claim 1, wherein the object is a software package.

17. The computer-implemented method of claim 1, wherein the object is a single YAML Ain't Markup Language (YAML)-based deployment.

18. The computer-implemented method of claim 1, wherein the object is an operator-based application bundle.

19. A computer program product, the computer program product comprising a non-transitory computer readable storage medium having program instructions embodied therewith, the program instructions executable by a computer to cause the computer to perform a method executed by a server in a cluster for checking an integrity of an object to be deployed to the cluster, the method comprising:

detecting a resource creation request from a client;

responsive to the resource creation request being an initial resource creation request for the object, verifying the integrity of the object based on properties in the resource creation request to create a release secret in the cluster for a positive integrity verification result for the object, the release secret representing a specific deployment configuration of the package on the cluster;

responsive to the resource creation request being other than the initial resource request, checking if the resource creation request corresponds to the specific deployment configuration of the object by checking against the release secret in the cluster; and

responsive to the resource creation request corresponding to a deployment of the object and the release secret being present in the cluster, creating a resource requested by the resource creation request in the cluster.

20. A server in a cluster for checking an integrity of an object to be deployed to the cluster, the server comprising:

a memory configured to store program code; and

a hardware processor configured to run the program code to:

detect a resource creation request from a client;

responsive to the resource creation request being an initial resource creation request for the object, verify the integrity of the package based on properties in the resource creation request to create a release secret in the cluster for a positive integrity verification result for the object, the release secret representing a specific deployment configuration of the object on the cluster;

responsive to the resource creation request being other than the initial resource request, check if the resource creation request corresponds to the specific deployment configuration of the object by checking against the release secret in the cluster; and

responsive to the resource creation request corresponding to a deployment of the package and the release secret being present in the cluster, create a resource requested by the resource creation request in the cluster.