

METHOD FOR MANAGING INSTITUTIONAL NODE AND DOMAIN CONTROLLER

TECHNICAL FIELD

[0001] One or more embodiments of this specification relate to the field of computer technologies, and in particular, to a method for managing an institutional node in a Kubernetes cluster, a domain controller, a computer-readable storage medium, and a computing device.

BACKGROUND

[0002] Secure multi-party privacy computation usually requires a plurality of participating institutional nodes to collaborate to complete joint computation. Management of these institutional nodes participating in joint computation is a prerequisite for subsequent secure multi-party privacy computation. In addition, emergence of Kubernetes (K8s for short) promotes popularization and implementation of popular technologies such as a microservices architecture, which makes development, operation and maintenance, and delivery increasingly simple. Increasingly more enterprises use Kubernetes as a container orchestration platform.

[0003] Therefore, in view of a feature that secure multi-party privacy computation requires a plurality of participating institutional nodes to collaborate to complete joint computation, embodiments of this specification provide a method for automatically managing an institutional node in secure multi-party privacy computation based on Kubernetes, so that it is simpler and more efficient to create and manage an institutional node.

SUMMARY

[0004] One or more embodiments of this specification describe a method for managing an institutional node in a Kubernetes cluster, and a domain controller, so that it is simpler and more efficient to create and manage an institutional node.

[0005] According to a first aspect, a method for managing a privacy computing node in a Kubernetes cluster is provided and performed by a domain controller. The domain controller

applies to the Kubernetes cluster for monitoring domain resource-related information in advance by using a monitoring mechanism. The method includes: obtaining a first resource change event based on the monitoring mechanism, where the first resource change event indicates that a custom domain resource is created in the Kubernetes cluster, content in the domain resource includes a
5 name of the domain resource and resource configuration information, and the name corresponds to a target institution; creating a namespace that has the name and that is unique to the target institution in the Kubernetes cluster; and creating another resource related to the target institution in the namespace based on the resource configuration information.

[0006] In an embodiment, the monitoring mechanism is a list-watch mechanism; and before
10 the obtaining a first resource change event, the method further includes: applying to the Kubernetes cluster for the monitoring domain resource-related information through a list-watch interface; and establishing a long connection to the Kubernetes cluster, to obtain a resource change event through the watch interface.

[0007] In an embodiment, the another resource includes a resource quota resource.

[0008] In an embodiment, the method further includes: obtaining a second resource change
15 event based on the monitoring mechanism, where the second resource change event indicates that the domain resource is updated; and correspondingly updating a related resource in the namespace based on content in an updated domain resource.

[0009] In an embodiment, the method further includes: obtaining a third resource change event
20 based on the monitoring mechanism, where the third resource change event indicates that the domain resource is deleted; and deleting the namespace and the another resource from the Kubernetes cluster in response to the third resource change event.

[0010] In an embodiment, the domain controller further applies to the Kubernetes cluster for
related information about a monitored computing node, and the method further includes: obtaining
25 a fourth resource change event based on the monitoring mechanism, where the fourth resource change event indicates that the target institution registers the computing node with the Kubernetes cluster; and adding status information of the registered computing node to the domain resource.

[0011] In an embodiment, the method further includes: obtaining a fifth resource change event
based on the monitoring mechanism, where the fifth resource change event indicates that the status
30 information of the registered computing node is changed; and correspondingly updating the status information in the domain resource.

[0012] According to a second aspect, a domain controller is provided. The domain controller applies to a Kubernetes cluster for monitoring domain resource-related information in advance by using a monitoring mechanism. The domain controller includes: a monitoring module, configured to obtain a first resource change event based on the monitoring mechanism, where the first resource
5 change event indicates that a custom domain resource is created in the Kubernetes cluster, content in the domain resource includes a name of the domain resource and resource configuration information, and the name corresponds to a target institution; a first creation module, configured to create a namespace that has the name and that is unique to the target institution in the Kubernetes cluster; and a second creation module, configured to create another resource related to the target
10 institution in the namespace based on the resource configuration information.

[0013] According to a third aspect, a computer-readable storage medium is provided. The computer-readable storage medium stores a computer program. When the computer program is executed in a computer, the computer is enabled to perform the method according to the first aspect.

[0014] According to a fourth aspect, a computing device is provided, and includes a memory
15 and a processor. The memory stores executable code. When the processor executes the executable code, the method according to the first aspect is implemented.

[0015] According to the method for managing an institutional node in a Kubernetes cluster and the domain controller that are disclosed in the embodiments of this specification, it can be simpler and more efficient to create and manage an institutional node, to facilitate subsequent
20 secure multi-party privacy computation.

BRIEF DESCRIPTION OF DRAWINGS

[0016] To describe the technical solutions in the embodiments of this specification more clearly, the following briefly describes the accompanying drawings needed for describing the embodiments. Clearly, the accompanying drawings in the following descriptions show merely
25 some embodiments of the present invention, and a person of ordinary skill in the art can still derive other drawings from these accompanying drawings without creative efforts.

[0017] FIG. 1 is a schematic diagram of a procedure of creating a namespace of an institutional node and a specific resource of the institutional node according to an embodiment of this specification;

- [0018] FIG. 2 is a schematic diagram of a procedure of updating a specific resource of an institutional node according to an embodiment of this specification;
- [0019] FIG. 3 is a schematic diagram of a procedure of deleting a namespace of an institutional node and a specific resource of the institutional node according to an embodiment of this specification;
- [0020] FIG. 4 is a schematic diagram of a procedure of adding status information of a computing node of an institution according to an embodiment of this specification;
- [0021] FIG. 5 is a schematic diagram of a procedure of updating status information of a computing node of an institution according to an embodiment of this specification; and
- [0022] FIG. 6 is a schematic diagram of a structural module of a domain controller according to an embodiment of this specification.

DESCRIPTION OF EMBODIMENTS

- [0023] The solutions provided in this specification are described below with reference to the accompanying drawings.
- [0024] As described above, management of institutional nodes participating in joint computation in K8s is a prerequisite for subsequent secure multi-party privacy computation. It should be understood that the institutional node is in a one-to-one correspondence with an institution, or the institutional node can be considered as a namespace (namespace) of an institution in K8s. Currently, in the industry, the namespace (namespace) and other resources corresponding to the institutional node are mainly created in a K8s cluster by manually executing a command line tool provided by K8s or by invoking, for a plurality of times, an API interface provided by K8s. However, this method has the following disadvantages:
- [0025] (1) It is inconvenient to maintain the namespace (namely, the namespace to which the institutional node belongs) of the institutional node. When there are increasingly more namespaces of institutional nodes in the cluster, a namespace that belongs to the institutional node and a namespace that does not belong to the institutional node cannot be effectively distinguished. Therefore, management is difficult.
- [0026] (2) In some cases, a quantity of resources (for example, pods) in the namespace of the institutional node needs to be limited. In this case, related resource quota (resource quota)

resources need to be created by manually executing the command line tool provided by K8s or by invoking the API interface provided by K8s.

[0027] (3) To view a status of a computing node of an institution, the command line tool provided by K8s needs to be manually executed or the API interface provided by K8s needs to be invoked. The operation is inconvenient.

[0028] Based on the foregoing observation and analysis, the embodiments of this specification provide a new solution. In this solution, based on K8s, a domain resource (Domain CRD) related to an institutional node is added, and a domain controller (Domain Controller) service module for managing the institutional node is added, to automatically manage an institutional node in secure multi-party privacy computation, so that it is simpler and more efficient to create and manage an institutional node, and view a node status.

[0029] FIG. 1 is a schematic diagram of a procedure of creating a namespace of an institutional node and a specific resource of the institutional node according to an embodiment of this specification. The figure relates to interaction between a user (for example, a K8s platform administrator), a K8s API server, and a domain controller.

[0030] It should be noted that the domain controller (Domain Controller) is a standard component that is newly disclosed in the embodiments of this specification and that does not belong to a K8s cluster. The domain controller can be designed to belong to the K8s cluster, or can be designed to be located outside the K8s cluster, and is communicatively connected to the K8s cluster.

[0031] The domain controller applies to the K8s cluster for monitoring domain resource-related information in advance by using a monitoring mechanism. In an embodiment, the monitoring mechanism can be a list-watch mechanism. Therefore, the domain controller can apply to the K8s cluster for the monitoring domain resource-related information through a list-watch interface. Further, the domain controller establishes a long connection to the K8s cluster, to obtain a resource change event through the watch interface. It can be understood that the monitoring mechanism is not limited to the list-watch mechanism, and can alternatively be another feasible monitoring mechanism.

[0032] As shown in FIG. 1, the procedure of creating a namespace and a related resource of an institutional node includes the following steps.

[0033] Step S110: A user creates a domain resource CR in the K8s cluster.

[0034] It should be noted that the domain resource does not belong to a resource type embedded in K8s, but belongs to an extended resource type. Specifically, K8s provides a manner of extending a resource type. One is a CRD whose full name is CustomResourceDefinition, that is, a custom resource, and Kubernetes CR represents a specific example (example) of the
5 Kubernetes CRD. Based on this, the domain resource example created in this specification can be referred to as a domain resource CR.

[0035] In an embodiment, the user creates a domain resource CR in the K8s cluster by using a K8s kubectl tool. In another embodiment, the user triggers an application to create a domain resource CR in the K8s cluster by invoking a K8s API.

10 **[0036]** In an embodiment, if an institution is registered through a platform, the platform assigns a unique institution name to the institution. The name corresponds to a domain resource name. Then, a platform administrator creates a domain resource CR for the institution in the K8s cluster through the platform. A specific example of a domain resource CR is displayed below, and content included in the domain resource is described with reference to this example:

```
15         1    apiVersion: kuscia.secretflow/v1
           2    kind: Domain
           3    metadata:
           4      name: alice-ab96242c2bc1a091
           5    spec:
20         6      cert: "xxxx"
           7      resourceQuota:
           8        podMaxCount: 100
           9      ...
          10    status:
25         11      nodeStatuses:
          12        - state: "Ready"
          13          id: "1673927061-1675394922047138439"
          14            name: "izuf6fpmmmb55do4m40ko2kz-kl6ofj-agent"
          15              ip: "11.0.0.2"
30         16      lastUpdateTime: 2023-02-04T04:24:23Z"
          17      ...
```

[0037] With reference to the foregoing example, it can be seen that the domain resource can include the following content:

- [0038] (1) Name (name): As shown in line 4, this represents a name of the domain resource CR. The domain controller creates a namespace having this name.
- [0039] (2) Resource quota (resourceQuota): Refer to lines 7–9. Line 8 represents a maximum quantity of pods in a resource quota CR. The domain controller creates a corresponding resource
5 quota CR based on description information of the resource quota.
- [0040] (3) Status (status): As shown in lines 10–17, this represents status information of all node resources of the institution. It should be understood that this is a function provided by the domain resource CR. This can make it more convenient for the platform to subsequently display node information of the institution or for black screen operation and maintenance.
- 10 [0041] Creation of the domain resource CR and the content included in the domain resource CR are described above.
- [0042] Step S120: The domain controller obtains a first resource change event based on the monitoring mechanism, where the first resource change event indicates that a custom domain resource CR is created in the Kubernetes cluster.
- 15 [0043] It should be noted that similar terms such as "first" in the first resource change event and "second" and "third" at other locations in this specification are used to distinguish between objects of the same type, and do not have other limitation functions such as ranking.
- [0044] Content in the domain resource CR includes a name of the domain resource and resource configuration information such as a resource quota (resource quota). The name of the
20 domain resource corresponds to a target institution. For details, refer to the foregoing code example.
- [0045] In an embodiment, the first resource change event includes the content in the domain resource CR. For example, the monitoring mechanism is a list-watch mechanism. Therefore, the domain controller can obtain the pushed first resource change event. The first resource change event includes the content in the domain resource CR.
- 25 [0046] Step S130: The domain controller creates a namespace that has the name and that is unique to the target institution in the Kubernetes cluster based on the name of the domain resource, and creates another resource related to the target institution in the namespace based on the resource configuration information.
- [0047] It should be noted that in a secure multi-party privacy computation scenario, it needs
30 to be ensured that private data of each computation participant or each institution is not out of a domain or is not disclosed. By creating a namespace unique to each institution and creating a

specific resource in the namespace, a resource isolation characteristic of the namespace can be implemented, so that each institution can securely participate in multi-party computation. In addition, different privacy computation nodes in the same institution namespace can effectively and collaboratively use internal resources.

5 **[0048]** For example, the resource configuration information includes description information of a resource quota (resourceQuota). Based on this, the domain controller can create a corresponding resource quota CR based on the description information.

[0049] The procedure of creating a namespace of an institutional node and a specific resource in the namespace is described above.

10 **[0050]** FIG. 2 is a schematic diagram of a procedure of updating a specific resource of an institutional node according to an embodiment of this specification. It should be understood that the update procedure shown in FIG. 2 can occur after the creation procedure shown in FIG. 1.

[0051] The update procedure shown in FIG. 2 includes the following steps.

[0052] Step S210: The user updates the domain resource CR in the K8s cluster.

15 **[0053]** In an embodiment, the user updates information in a spec field in the domain resource CR. For example, resource quota (resourceQuota) information in the domain resource CR is updated. For example, "podMaxCount: 100" in line 8 in the foregoing example is changed to "podMaxCount: 80".

[0054] Step S220: The domain controller obtains a second resource change event based on the
20 monitoring mechanism, where the second resource change event indicates that the domain resource is updated.

[0055] In an embodiment, the second resource change event includes updated content in the domain resource CR. For example, the monitoring mechanism is a list-watch mechanism. Therefore, the domain controller can obtain the pushed second resource change event. The second
25 resource change event includes the updated content in the domain resource CR.

[0056] Step S230: The domain controller correspondingly updates a related resource based on content in an updated domain resource.

[0057] In an embodiment, the resource quota (resourceQuota) information in the domain resource CR is changed. Therefore, a resource quota resource can be correspondingly modified.

30 **[0058]** The procedure of updating a specific resource in a namespace of an institutional node is described above.

- [0059] FIG. 3 is a schematic diagram of a procedure of deleting a namespace of an institutional node and a specific resource of the institutional node according to an embodiment of this specification. It should be understood that the deletion procedure shown in FIG. 3 can occur after the creation procedure shown in FIG. 1 or the update procedure shown in FIG. 2.
- 5 [0060] The deletion procedure shown in FIG. 3 includes the following steps.
- [0061] Step S310: The user deletes the domain resource CR from the K8s cluster.
- [0062] Step S320: The domain controller obtains a third resource change event based on the monitoring mechanism, where the third resource change event indicates that the domain resource is deleted.
- 10 [0063] For example, the monitoring mechanism is a list-watch mechanism. Therefore, the domain controller can obtain the pushed third resource change event.
- [0064] Step S330: The domain controller deletes a corresponding namespace and a resource in the namespace from the Kubernetes cluster.
- [0065] The procedure of deleting a namespace of an institutional node and a resource in the namespace is described above.
- 15 [0066] FIG. 4 is a schematic diagram of a procedure of adding status information of a computing node of an institution according to an embodiment of this specification. It should be noted that the domain controller further applies to the K8s cluster for related information about a monitored node in advance by using the monitoring mechanism. The node information addition procedure shown in FIG. 4 can occur after the procedure shown in FIG. 1 or FIG. 2.
- 20 [0067] The node information addition procedure shown in FIG. 4 includes the following steps.
- [0068] Step S410: The domain controller obtains a fourth resource change event based on the monitoring mechanism, where the fourth resource change event indicates that the target institution registers the privacy computing node with the Kubernetes cluster.
- 25 [0069] In an embodiment, the fourth resource change event includes status information of the privacy computing node. For example, the monitoring mechanism is a list-watch mechanism. Therefore, the domain controller can obtain the pushed fourth resource change event. The fourth resource change event includes the status information of the privacy computing node.
- [0070] Step S420: The domain controller adds the status information of the privacy computing node to the domain resource CR.
- 30 [0071] For example, for the status information added to the domain resource CR, refer to lines

10–17 in the foregoing code example.

[0072] The procedure of adding status information of a computing node of an institution is described above.

5 [0073] FIG. 5 is a schematic diagram of a procedure of updating status information of a computing node of an institution according to an embodiment of this specification. The node information update procedure shown in FIG. 5 can occur after the node information addition procedure shown in FIG. 4.

[0074] The node information update procedure shown in FIG. 5 includes the following steps.

10 [0075] Step S510: The domain controller obtains a fifth resource change event based on the monitoring mechanism, where the fifth resource change event indicates that the status information of the privacy computing node is changed.

[0076] In an embodiment, the fifth resource change event includes updated status information of the privacy computing node. For example, the monitoring mechanism is a list-watch mechanism. Therefore, the domain controller can obtain the pushed fifth resource change event. The fifth
15 resource change event includes the updated status information of the privacy computing node.

[0077] In addition, in an embodiment, the changed status information includes a node status, for example, line 12 in the foregoing code example.

[0078] Step S520: The domain controller correspondingly updates the status information in the domain resource.

20 [0079] The procedure of updating status information of a computing node of an institution is described above.

[0080] In conclusion, according to the method for managing a privacy computing node disclosed in the embodiments of this specification, it can be simpler and more efficient to create and manage an institutional node. Details are as follows:

25 [0081] (1) A domain resource (Domain CRD) and a domain controller (Domain Controller) service module are added, to implement automatic management such as creation/update/deletion of a namespace of an institutional node, so that a K8s-related command does not need to be executed in a plurality of steps to create resources such as the namespace (namespace)/a resource quota (resource quota) of the institutional node.

30 [0082] (2) The domain resource (Domain CRD) can store a summary of some other basic information needed to create an institution. Automatic management of related resources can be

implemented by using the domain controller (Domain Controller).

[0083] (3) The domain controller (Domain Controller) updates statuses of computing nodes (nodes) of the institution to the domain resource (Domain CRD), so that it is more convenient for the platform to subsequently display node information of the institution or for black screen operation and maintenance.

[0084] Corresponding to the foregoing management method, an embodiment of this specification further provides a domain controller. FIG. 6 is a schematic diagram of a structural module of a domain controller according to an embodiment of this specification. The domain controller applies to a Kubernetes cluster for monitoring domain resource-related information in advance by using a monitoring mechanism. As shown in FIG. 6, the domain controller 600 includes:

a first monitoring module 601, configured to obtain a first resource change event based on the monitoring mechanism, where the first resource change event indicates that a custom domain resource is created in the Kubernetes cluster, content in the domain resource includes a name of the domain resource and resource configuration information, and the name corresponds to a target institution; a first creation module 603, configured to create a namespace that has the name and that is unique to the target institution in the Kubernetes cluster; and a second creation module 605, configured to create another resource related to the target institution in the namespace based on the resource configuration information.

[0085] In an embodiment, the another resource includes a resource quota resource.

[0086] In an embodiment, the domain controller 600 further includes: a second monitoring module 607, configured to obtain a second resource change event based on the monitoring mechanism, where the second resource change event indicates that the domain resource is updated; and a first update module 609, configured to correspondingly update a related resource in the namespace based on content in an updated domain resource.

[0087] In an embodiment, the domain controller 600 further includes: a third monitoring module 611, configured to obtain a third resource change event based on the monitoring mechanism, where the third resource change event indicates that the domain resource is deleted; and a deletion module 613, configured to delete the namespace and the another resource from the Kubernetes cluster in response to the third resource change event.

[0088] In an embodiment, the domain controller 600 further applies to the Kubernetes cluster for related information about a monitored computing node, and the domain controller 600 further includes: a fourth monitoring module 615, configured to obtain a fourth resource change event

based on the monitoring mechanism, where the fourth resource change event indicates that the target institution registers the registered computing node with the Kubernetes cluster; and an addition module 617, configured to add status information of the registered computing node to the domain resource.

5 **[0089]** In a specific embodiment, the domain controller 600 further includes: a fifth monitoring module 619, configured to obtain a fifth resource change event based on the monitoring mechanism, where the fifth resource change event indicates that the status information of the registered computing node is changed; and a second update module 621, configured to correspondingly update the status information in the domain resource.

10 **[0090]** According to an embodiment in still another aspect, a computer-readable storage medium is further provided. The computer-readable storage medium stores a computer program. When the computer program is executed in a computer, the computer is enabled to perform the method described with reference to FIG. 1, FIG. 2, FIG. 3, or FIG. 4.

15 **[0091]** According to an embodiment in yet another aspect, a computing device is further provided, and includes a memory and a processor. The memory stores executable code. When the processor executes the executable code, the method described with reference to FIG. 1, FIG. 2, FIG. 3, or FIG. 4 is implemented.

20 **[0092]** A person skilled in the art should be aware that in the foregoing one or more examples, functions described in the present invention can be implemented by hardware, software, firmware, or any combination thereof. When being implemented by software, these functions can be stored in a computer-readable medium or transmitted as one or more instructions or code on a computer-readable medium.

25 **[0093]** The objectives, technical solutions, and beneficial effects of the present invention are further described in detail in the foregoing specific implementations. It should be understood that the foregoing descriptions are merely specific implementations of the present invention, but are not intended to limit the protection scope of the present invention. Any modification, equivalent replacement, improvement, and the like made based on the technical solutions of the present invention shall fall within the protection scope of the present invention.

CLAIMS

What is claimed is:

1. A method for managing an institutional node in a Kubernetes cluster, performed by a domain controller, wherein the domain controller applies to the Kubernetes cluster for monitoring
5 domain resource-related information in advance by using a monitoring mechanism, and the method comprises:

obtaining a first resource change event based on the monitoring mechanism, wherein the first resource change event indicates that a custom domain resource is created in the Kubernetes cluster, content in the domain resource comprises a name of the domain resource and resource
10 configuration information, and the name corresponds to a target institution;

creating a namespace that has the name and that is unique to the target institution in the Kubernetes cluster; and

creating another resource related to the target institution in the namespace based on the resource configuration information.

15 2. The method according to claim 1, wherein the monitoring mechanism is a list-watch mechanism; and before the obtaining a first resource change event, the method further comprises:

applying to the Kubernetes cluster for the monitoring domain resource-related information through a list-watch interface; and

establishing a long connection to the Kubernetes cluster, to obtain a resource change event
20 through the watch interface.

3. The method according to claim 1, wherein the another resource comprises a resource quota resource.

4. The method according to claim 1, further comprising:

obtaining a second resource change event based on the monitoring mechanism, wherein the
25 second resource change event indicates that the domain resource is updated; and

correspondingly updating a related resource in the namespace based on content in an updated domain resource.

5. The method according to claim 1, further comprising:

obtaining a third resource change event based on the monitoring mechanism, wherein the
30 third resource change event indicates that the domain resource is deleted; and

deleting the namespace and the another resource from the Kubernetes cluster in response to the third resource change event.

6. The method according to claim 1, wherein the domain controller further applies to the Kubernetes cluster for related information about a monitored computing node, and the method further comprises:

obtaining a fourth resource change event based on the monitoring mechanism, wherein the fourth resource change event indicates that the target institution registers the computing node with the Kubernetes cluster; and

adding status information of the registered computing node to the domain resource.

7. The method according to claim 6, further comprising:

obtaining a fifth resource change event based on the monitoring mechanism, wherein the fifth resource change event indicates that the status information of the registered computing node is changed; and

correspondingly updating the status information in the domain resource.

8. A domain controller, wherein the domain controller applies to a Kubernetes cluster for monitoring domain resource-related information in advance by using a monitoring mechanism, and the domain controller comprises:

a monitoring module, configured to obtain a first resource change event based on the monitoring mechanism, wherein the first resource change event indicates that a custom domain resource is created in the Kubernetes cluster, content in the domain resource comprises a name of the domain resource and resource configuration information, and the name corresponds to a target institution;

a first creation module, configured to create a namespace that has the name and that is unique to the target institution in the Kubernetes cluster; and

a second creation module, configured to create another resource related to the target institution in the namespace based on the resource configuration information.

9. A computer-readable storage medium, wherein the computer-readable storage medium stores a computer program, and when the computer program is executed in a computer, the computer is enabled to perform the method according to any one of claims 1 to 7.

10. A computing device, comprising a memory and a processor, wherein the memory stores executable code, and when the processor executes the executable code, the method according to

any one of claims 1 to 7 is implemented.