

Claims:

1. A computer-implemented method, comprising:
receiving a task definition, the task definition indicating at least:
a location at which a software image is located; and
information usable to determine an amount of a resource to allocate to a software container for the software images, the software container being a user space instance that runs in isolation from other software containers running on a same virtual machine instance;
determining, based at least in part on a placement scheme and available resources in a set of virtual machine instances, a virtual machine instance from the set of virtual machine instances in which to launch the software container at least by parsing, via a scheduler application, the task definition for information that indicates an affinity or anti-affinity for certain types of virtual machine instances; and
causing the software image to be obtained from the location indicated by the task definition and launched in the software container within the virtual machine instance, wherein the software container is allocated the amount of the resource.
2. The computer-implemented method of claim 1, wherein the set of virtual machine instances are one or more virtual machine instances that have been registered in association with each other as a cluster.
3. The computer-implemented method of claim 1, further comprising receiving, via an application program interface, a request to instantiate the task definition, wherein the, wherein the application programming interface is provided by a computing resource service provider.
4. The computer-implemented method of claim 1, wherein causing the software image to be launched further comprises executing a scheduler specified by a customer of a computing resource service provider, the scheduler configured to select the virtual machine instance from a set of virtual machine instances to run the software containers.

5. The computer-implemented method of claim 1, wherein the information that indicates an affinity or anti-affinity for certain types of virtual machine instances is a tag.

6. The computer-implemented method of claim 1, wherein:
the virtual machine instance is associated with a customer of a computing resource service provider; and
the task definition is defined by the customer through an editing interface of a template service provided by the computing resource service provider.

7. A system, comprising:
one or more processors; and
memory including instructions that, as a result of execution by the one or more processors, cause the system to:
receive a task definition, the task definition indicating at least:
a location at which a software image is located; and
information usable to determine an amount of a resource to allocate to a software container for the software image, the software container:
being an isolated user space instance that runs in isolation from other software containers running on a virtual machine instance; and
having a share of resources of the virtual machine instance;
determine, based at least in part on a placement scheme and available resources in a set of virtual machine instances, the virtual machine instance from the set of virtual machine instances in which to launch the software container at least by causing the system to parse the task definition for information that indicates an affinity or anti-affinity for certain types of virtual machine instances; and
launch the software image, obtained from the location indicated by the task definition, in the software container within the virtual machine instance, the software container being allocated the share of the resource in accordance with the amount of the resource indicated by the task definition.

8. The system of claim 7, wherein the set of virtual machine instances operate within a virtual network of a customer of a computing resource service provider.

9. The system of claim 7, wherein the set of virtual machine instances are associated with an auto-scaling group of an auto-scaling service that causes virtual machine instances to be added to or removed from the auto-scaling group depending at least in part on an amount of available resources determined from the information included in the task definition.

10. The system of claim 7, wherein:
the software container is a second software container; and
the task definition is received as a result of a first software container encountering an error in execution of the software image.

11. The system of claim 7, wherein the instructions that cause the system to determine the set of virtual machine instances, further include instructions that cause the system to determine the virtual machine instance further based at least in part according to a load-balancing distribution scheme of a load-balancer service.

12. The system of claim 7, wherein the instructions further include instructions that cause the system to:
aggregate metrics about the software container to form aggregated metrics; and
allow customers of a computing resource service provider hosting the system to monitor resource utilization of the software container at least by providing the aggregated metrics to a resource monitoring service of the computing resource service provider.

13. The system of claim 12, wherein the metrics include one or more of:
an amount of memory used by the software container,
a number of times that an error in execution of the software image occurred, or
processor usage by the software container.

14. A non-transitory computer-readable storage medium having stored thereon executable instructions that, as a result of being executed by one or more processors of a computer system, cause the computer system to at least:

receive a task definition, the task definition indicating at least:
a location at which a software image is located; and

information usable to determine an amount of a computing resource to allocate to a software container for the software image, the software container having a share of a resource of a computer system instance;

determine, based at least in part on a placement scheme and available resources in a set of computer system instances, the computer system instance from the set of computer system instances in which to launch the software container at least by at least by causing the computer system to parse the task definition for information that indicates an affinity or anti-affinity for certain types of virtual machine instances; and

launch the software image, obtained from the location indicated by the task definition, as the software container within the computer system instance, the software container being allocated the share of the resource from available resources of the computer system instance in accordance with the amount indicated by the task definition.

15. The non-transitory computer-readable storage medium of claim 14, wherein the location from which the software image is located is a location in a data store of a computing resource service provider that hosts the computer system.

16. The non-transitory computer-readable storage medium of claim 14, wherein the executable instructions further include instructions that cause the computer system to:

aggregate metrics about the software container to form aggregated metrics; and
provide the aggregated metrics to a resource monitoring service of a computing resource service provider hosting the computer system.

17. The non-transitory computer-readable storage medium of claim 16, wherein the metrics aggregated include one or more of:

an amount of time that the one or more processors spent executing a system call made by the software container,

a number of read and write operations performed by the software container, or

a number of input/output operations queued for the software container.

18. The non-transitory computer-readable storage medium of claim 16, wherein:

the set of computer system instances are associated with a customer of a computing resource service provider that hosts the computer system; and

the aggregated metrics meet a predefined condition that causes the resource monitoring service to trigger an alarm that notifies the customer that the predefined condition has been met.

19. The non-transitory computer-readable storage medium of claim 18, wherein the predefined condition is at least one of:

an indication of an occurrence of an error state of the software container, or
that at least one of the aggregated metrics has reached a value relative to a threshold.

20. The non-transitory computer-readable storage medium of claim 18, wherein the alarm that notifies the customer includes one or more of:

a text message sent to a device associated with the customer,
an e-mail sent to an e-mail location associated with the customer, or
a visual status indicator displayed on a user interface accessible to the customer.

21. The computer-implemented method of claim 1, wherein:
the software container is a second software container; and
the task definition is received as a result of a first software container encountering an error in execution of the software image.