

CLAIMS

1. A computer-implemented method, comprising:

receiving, from a customer of a computing resource service provider, one or more application programming interface (API) calls to create a cluster of virtual machine instances running a process; and

in response to the one or more API calls:

instantiating a plurality of virtual machine instances;

associating the plurality of virtual machine instances with a cluster identifier; and

for each virtual machine instance of the plurality of virtual machine instances:

launching a software container in the virtual machine instance; and

executing at least a portion of the process within the software container.

2. The computer-implemented method of claim 1, wherein the software container is a user space instance that runs in isolation from another software container running in the virtual machine instance.

3. The computer-implemented method of claim 1, wherein the cluster is associated with a plurality of instances that includes the instance.

4. The computer-implemented method of claim 1, wherein instantiating the plurality of virtual machine instances further includes instantiating the plurality of virtual machine instances within a virtual private cloud.

5. The computer-implemented method of claim 1, wherein the software container is linked to another software container in a manner specified by a task definition.

6. The computer-implemented method of claim 1, wherein the computer-implemented method further comprises:

obtaining a task definition that:

indicates a location of a software image associated with the process;

and

includes information usable in determining an amount of a computing resource of the virtual machine instance to allocate to the software container; and further for each virtual machine instance of the plurality of virtual machine instances:

5 obtaining the software image from the location;
 allocating the amount of the computing resource of the virtual machine instance to the software container; and
 executing the at least a portion of the process based at least in part on the software image.

10 7. The computer-implemented method of claim 6, wherein the software image is data that represents a state of the software container at a time of creation of the software image.

 8. A system, comprising:
 one or more processors; and
15 memory including executable instructions that, if executed by the one or more processors, cause the system to:

 obtain a task definition, the task definition indicating a name of a task to perform, a location of a container image for the task, and a port for the task;

 receive a request to run, within a cluster of virtual machine instances,
20 one or more containers to perform the task, wherein the request further indicates an amount of a resource to be allocated to the task;

 select, via a scheduler, an available virtual machine instance from the cluster in which to run the one or more containers; and

 run, within the available virtual machine instance and in accordance
25 with the task definition, the one or more containers based at least in part on the container image at least by causing the system to:

 obtain, based at least in part on the task definition, the container image from the location;

 allocate the amount of the resource to the one or more
30 containers;

 assign the port to the one or more containers; and
 run the one or more containers to perform the task.

9. The system of claim 8, wherein the amount of the resource is a portion of a quantity of the resource allocated to the available virtual machine instance.

10. The system of claim 8, wherein:

the task definition further includes affinity information than indicates an
5 affinity or anti-affinity for certain types of virtual machine instances; and

the executable instructions that cause the system to select the available virtual machine instance further includes instructions that cause the system to select the available virtual machine instance based at least in part on the affinity information.

11. The system of claim 8, wherein a load balancer balances traffic
10 across virtual machine instances of the cluster.

12. The system of claim 8, wherein a Docker container engine manages the one or more containers running in the available virtual machine instance.

13. The system of claim 8, wherein the task definition further specifies that:

15 the one or more containers are assigned to a group; and
members of the group are coordinated to run on a same virtual machine instance.

14. The system of claim 8, wherein the executable instructions further include instructions that cause the system to:

20 obtain metrics associated with the one or more containers running within the available virtual machine instance; and
provide the metrics to a resource monitoring service.

15. The system of claim 14, wherein the metrics include at least one of:

processor usage by the one or more containers,
25 processor time used by the one or more containers,
amount of memory used by the one or more containers, or
amount of storage space used by the one or more containers.

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

as a result of a task being scheduled to be performed in an instance of a cluster of instances according to a task definition, receive a communication from a scheduler to perform the task;

in response to receipt of the communication, instantiate a container in the instance to perform the task;

monitor the container to obtain health information about the container; and

as a result of a determination, made based at least in part on the health information, that the container has experienced an error, instantiate a new container to replace the container that experienced the error.

17. The non-transitory computer-readable storage medium of claim 16, wherein a quantity of instances of the cluster is dynamically increased or decreased, depending at least in part on resource demand by containers in the instances of the cluster, by an auto-scaling service.

18. The non-transitory computer-readable storage medium of claim 16, wherein the instance of the cluster is located within a virtual private cloud.

19. The non-transitory computer-readable storage medium of claim 16, wherein the health information includes at least one of:

a heartbeat of the container,

a lifespan of the container,

a status of the container, or

an occurrence of an error with the container.

20. The non-transitory computer-readable storage medium of claim 16, wherein the executable instructions are executed in the instance to operate an agent that starts and runs containers scheduled on the instance.