

WHAT IS CLAIMED IS:

- 1 1. A system comprising:
2 a host machine configured to run a container orchestrator, the host machine
3 comprising:
4 a database comprising a multi-dimensional data store configured to
5 define a network data model, wherein the network data model is
6 configured to support one or more network data attributes, the
7 database further comprising a network configuration;
8 a processor; and
9 non-transitory computer readable media comprising instructions
10 executable by the processor to:
11 obtain, via the database, the network configuration;
12 spawn a container according to the network configuration,
13 wherein the container is configured, based on the network
14 configuration, to be coupled to a network overlay via a
15 network interface;
16 receive, via the network interface, incoming data associated
17 with the container, the incoming data having attached one
18 or more attached network data attributes; and
19 identify, via the database, the attached one or more network
20 data attributes attached to the incoming data as one or more
21 network data attributes of the network data model.
- 1 2. The system of claim 1, wherein the instructions are further executable
2 by the processor to:
3 attach one or more network data attributes associate to outgoing data, wherein
4 outgoing data is transmitted by the container and the one or more data
5 attributes attached to the outgoing data are based, at least in part, on the
6 container; and
7 transmit, via the network interface, the outgoing data.
- 1 3. The system of claim 1, wherein the instructions are further executable
2 by the processor to:

3 determine whether the container is communicating with a second container
4 within a shared pod, wherein the shared pod comprises one or more
5 containers including the container and the second container on the host
6 machine; and
7 for intrapod outgoing data transmitted by the container to the second
8 container, transmit a memory pointer associated with the location of the
9 intrapod outgoing data.

1 4. The system of claim 3, wherein the instructions are further executable
2 by the processor to:
3 for intrapod incoming data, receive the memory pointer associated with the
4 location of the intrapod incoming data.

1 5. The system of claim 1, wherein the instructions are further executable
2 by the processor to:
3 add, via the database, an additional attribute to the one or more network data
4 attributes of the network data model.

1 6. The system of claim 1, wherein the instructions are further executable
2 by the processor to:
3 remove, via the database, at least one of the one or more network data
4 attributes from the network data model.

1 7. The system of claim 1, wherein the instructions are further executable
2 by the processor to:
3 prioritize forwarding of the incoming data based, at least in part, on the one or
4 more attached network data attributes.

1 8. The system of claim 1, wherein the one or more network data attributes
2 includes at least a quality of service, security, routing, switching, proof of origin,
3 proof of delivery, and packet behavior monitoring associated with data received and
4 transmitted by the container.

1 9. An apparatus comprising:
2 a processor; and

3 non-transitory computer readable media comprising instructions executable by
4 the processor to:
5 obtain, via a database, a network configuration;
6 spawn a container according to the network configuration, wherein the
7 container is configured, based on the network configuration, to be
8 coupled to a network overlay via a network interface;
9 receive, via the network interface, incoming data associated with the
10 container, the incoming data having attached one or more attached
11 network data attributes; and
12 identify, via the database, the attached one or more network data
13 attributes attached to the incoming data as one or more network
14 data attributes of the network data model,
15 wherein the database comprises a multi-dimensional data store
16 configured to define a network data model, wherein the network
17 data model is configured to support one or more network data
18 attributes.

1 10. The apparatus of claim 9, wherein the instructions are further
2 executable by the processor to:
3 attach one or more network data attributes associate to outgoing data, wherein
4 outgoing data is transmitted by the container and the one or more data
5 attributes attached to the outgoing data are based, at least in part, on the
6 container; and
7 transmit, via the network interface, the outgoing data.

1 11. The apparatus of claim 9, wherein the instructions are further
2 executable by the processor to:
3 determine whether the container is communicating with a second container
4 within a shared pod, wherein the shared pod comprises one or more
5 containers including the container and the second container on the host
6 machine; and

7 for intrapod outgoing data transmitted by the container to the second
8 container, transmit a memory pointer associated with the location of the
9 intrapod outgoing data.

1 12. The apparatus of claim 11, wherein the instructions are further
2 executable by the processor to:
3 for intrapod incoming data, receive the memory pointer associated with the
4 location of the intrapod incoming data.

1 13. The apparatus of claim 9, wherein the instructions are further
2 executable by the processor to:
3 add, via the database, an additional attribute to the one or more network data
4 attributes of the network data model.

1 14. The apparatus of claim 9, wherein the instructions are further
2 executable by the processor to:
3 remove, via the database, at least one of the one or more network data
4 attributes from the network data model.

1 15. The apparatus of claim 9, wherein the instructions are further
2 executable by the processor to:
3 prioritize forwarding of the incoming data based, at least in part, on the one or
4 more attached network data attributes.

1 16. A method comprising:
2 defining, at an edge database, a network data model configured to support one
3 or more network data attributes;
4 providing, via the edge database, a network configuration;
5 obtaining, via an orchestrator, the network configuration from the edge
6 database;
7 spawning, via the orchestrator, a container according to the network
8 configuration, wherein spawning the container further comprises coupling
9 the container, based on the network configuration, to a network overlay via
10 a network interface;

11 receiving, via the network interface, incoming data associated with the
12 container, the incoming data having attached one or more attached
13 network data attributes; and
14 identifying, via the database, the attached one or more network data attributes
15 attached to the incoming data as one or more network data attributes of the
16 network data model.

1 17. The method of claim 16, further comprising:
2 determining whether the container is communicating with a second container
3 within a shared pod, wherein the shared pod comprises one or more
4 containers including the container and the second container on the host
5 machine; and
6 for intrapod outgoing data transmitted by the container to the second
7 container, transmitting a memory pointer associated with the location of
8 the intrapod outgoing data.

1 18. The method of claim 17, further comprising:
2 for intrapod incoming data, receiving the memory pointer associated with the
3 location of the intrapod incoming data.

1 19. The method of claim 16 further comprising:
2 add, via the database, an additional attribute to the one or more network data
3 attributes of the network data model; and
1 removing, via the database, at least one of the one or more network data
2 attributes from the network data model.

1 20. The method of claim 16 further comprising:
2 prioritizing forwarding of the incoming data based, at least in part, on the one
3 or more attached network data attributes.