This document presents an example HOT including policy according to (https://wiki.openstack.org/wiki/Heat/PolicyExtension). There is a companion drawing (https://docs.google.com/drawings/d/1nridrUUwNaDrHQoGwSJ_KXYC7ik09wUuV3vXw1MyvlY) that shows the stack being prescribed.

The stack in question is a simple 3-tier web application with high availability. The first tier is a pair of HTTP servers, which handle simple requests and delegate complex processing to the next tier. The HTTP servers must go on distinct racks. That next tier has four application servers, prescribed to be spread in a slightly sophisticated way. The appservers must be spread across at least two racks, with no more than two appservers on a single rack. Further, no two appservers can be on the same host (Physical Machine). The appservers, like the HTTP servers, are stateless and do not communicate with each other. All persistence is done in the final tier, which holds a symmetrically mirrored pair of database servers. Each database server has a Cinder volume that is hosted on a locally attached disk. The database servers must also be spread across racks. The allowed network communication is: between the Internet and the HTTP servers, between the HTTP servers and the appservers, between the appservers and the DB servers, and among the DB servers.

A separate line of inquiry is debating how to refer to pre-existing networky things. I do not want to pre-judge the outcome there. For now I will just use the UUID 11111111-2222-3333-4444-555555555555 to represent the Internet.

Without further ado, the template:

```
heat_template_version: 2013-05-23
description: 3 tier web app - 2,2x2,2
resources:
      http-server-1:
            type: OS::Nova::Server
            properties:
                  user_data: (elided for brevity)
                  key_name: Francis
                  flavor: m1.medium
                  image: http-server-image
      http-server-2:
            type: OS::Nova::Server
            properties:
                  user_data: (elided for brevity)
                  key_name: Francis
                  flavor: m1.medium
                  image: http-server-image
      app-server-11:
            type: OS::Nova::Server
```

```
      properties:
            user_data: (elided for brevity)
            key_name: Francis
            flavor: m1.large
            image: app-server-image
app-server-12:
      type: OS::Nova::Server
      properties:
            user_data: (elided for brevity)
            key_name: Francis
            flavor: m1.large
            image: app-server-image
app-server-21:
      type: OS::Nova::Server
      properties:
            user_data: (elided for brevity)
            key_name: Francis
            flavor: m1.large
            image: app-server-image
app-server-22:
      type: OS::Nova::Server
      properties:
            user_data: (elided for brevity)
            key_name: Francis
            flavor: m1.large
            image: app-server-image
db-server-1:
      type: OS::Nova::Server
      properties:
            user_data: (elided for brevity)
            key_name: Francis
            flavor: m1.xlarge
            image: app-server-image
db-vol-1:
      type: AWS::EC2::Volume
      properties:
            Size: 250
db-vol-1-attach:
      type: AWS::EC2::VolumeAttachment
      properties:
            Device: /dev/vdb
            InstanceID: { get_resource : db-server-1 }
            VolumeID: { get_resource : db-vol-1 }
```

```
          policies:
             -
               type: OS::NetMaxHops
               properties: { hops: 0 }
     db-server-2:
          type: OS::Nova::Server
          properties:
               user_data: (elided for brevity)
               key_name: Francis
               flavor: m1.xlarge
               image: app-server-image
     db-vol-2:
          type: AWS::EC2::Volume
          properties:
               Size: 250
     db-vol-2-attach:
          type: AWS::EC2::VolumeAttachment
          properties:
               Device: /dev/vdb
               InstanceID: { get_resource : db-server-2 }
               VolumeID: { get_resource : db-vol-2 }
          policies:
             -
               type: OS::NetMaxHops
               properties: { hops: 0 }
groups:
     id: alluvit
     members:
        -
          id: http-tier
          members:
               - { get_resource: http-server-1 }
               - { get_resource: http-server-2 }
          policies:
             -
               type: AntiCollocate
               properties: { level: rack }
          relationships:
             -
               related: 11111111-2222-3333-4444-555555555555
               policies:
                    - { type: NetworkReachability }
        -
```

```
  id: app-tier
  members:
     -
       id: app-tier-1
       members:
            - { get_resource: app-server-11 }
            - { get_resource: app-server-12 }
       policies:
          -
            type: AntiCollocate
            properties: { level: rack }
     -
       id: app-tier-2
       members:
            - { get_resource: app-server-21 }
            - { get_resource: app-server-22 }
       policies:
          -
            type: AntiCollocate
            properties: { level: rack }
  policies: # of app-tier
     -
       type: AntiCollocate
       properties: { level: PM }
  relationships: # of app-tier
     -
       related: http-tier
       policies:
            - { type: NetworkReachability }
-
  id: db-tier
  members:
     -
       id: db-tier-1
       members:
            - {get_resource: db-server-1}
            - {get_resource: db-vol-1}
            - {get_resource: db-vol-1-attach}
     -
       id: db-tier-2
       members:
            - {get_resource: db-server-2}
            - {get_resource: db-vol-2}
```

```
          - {get_resource: db-vol-2-attach}
policies: # of db-tier
   -
     type: AntiCollocation
     properties: {level: rack}
   -
     type: NetworkReachability
relationships: # of db-tier
   -
     related: app-tier
     policies:
          - {type: NetworkReachability}
```