**Scaling Application in Production:**

To scale the application running in Production we can have multiple tomcat servers (hosting .war file) running in separate Docker containers. In frontend of those servers, there will be a single Nginx server (hosting static content). That Nginx server will reverse proxy to the tomcat servers and will load balance using a round-robin methodology.

**Desired Architecture:**

**Docker Containers**

Tomcat(App)

Tomcat(App)

Tomcat(App)

Tomcat(App)

Browser

**VIRTUAL MACHINE (aws, virtualbox)**

**HOST MACHINE (Workstation)**

In our current swarm infrastructure we are putting a tomcat container hosting our dynamic application components behind a Nginx proxy container hosting the static components of the application. So any request for static content is proxied via nginx on the frontend to tomcat running as backend.

Once the application components are deployed as services to a swarm, we can use the Docker CLI to scale the number of containers in the service. So we can the scale the number of tomcat containers and the load balancing between the instances will be taken care by Nginx default algorithm.

Any number of Tomcat instances can be configured for load-balancing using NGINX's default round-robin algorithm. Load-balancing is configured through the “default.conf” file, in the upstream configuration section:

upstream backend {

app\_ip1:8080;

app\_ip2:8080;

app\_ip3:8080;

}

Hardware requirements:

As we scale up the application in Production we would require more servers to be joined as swarm nodes with swarm manager for appropriate load handling.