**Installing and Configuring the DNS Server (5p)**

Let's say that you have purchased a domain name your\_domain\_name (think up it!) for your company and want to set up your DNS server, instead of using DNS hosting services.

1. Go to the official repository of the world's most popular DNS server on docker-hub: <https://hub.docker.com/r/ubuntu/bind9>. In the Setup Instructions section, you need to find the command to create an authoritative DNS server container.

Error:

Error starting userland proxy: listen tcp4 0.0.0.0:port: bind: address already in use.

means that this port is already occupied by some program (you can view the list of occupied ports using the netstat -46anep command). The systemd-resolve system service cannot be stopped (otherwise the Internet access will turn off), so you will have to disable the ability to connect to your server from the outside: remove the port forwarding options from docker run command (they start with --publish)

Before re-creating the container, you need to delete the old one with the command:   
docker rm bind9

To view a list of all containers (and their IDs), use the command:

docker ps -a

2. The correct way to change files inside the volume container are either editing files by the program from inside the container, or editing them from the outside and copying them with the docker cp command, or using bind mount. The latter method is the most convenient when you need to frequently change files while configuring the container. Copy the /etc/bind folder with settings from the internal volume to the etc\_bind folder in the current directory and replace volume with bind mount when recreating the container:

docker cp bind9:/etc/bind etc\_bind

docker rm -f bind9

repeat the container creation command, where instead of --volume /etc/bind

specify the option --volume "`pwd`/etc\_bind":/etc/bind

3. Create a copy “your\_domain\_name” of the file “db.local” in the etc\_bind folder with a description of your new zone (the zone name must contain your name). In the SOA zone header, replace localhost by the name of your zone, and root.localhost by your email address (instead of @ here is a dot). In the remaining entries, correct 127.0.0.1 to your imaginary server's IP address, and localhost to the name of your zone. Add a couple of computers with fake IP addresses to your zone:

ComputerName1 IN A IP address1

ComputerName2 IN A IP address2

4. Add the new zone file to the list of DNS server zones in the file

etc\_bind/named. conf.default-zones

5. Restart the container with the command

docker container restart bind9

and make a test request to your DNS server:

nslookup -port=DNS\_server\_port ComputerName1.your\_domain\_name DNS\_server\_IP

From your docker host, you need to specify the server port 53, DNS\_server\_IP=internal IP in the docker network. You can find it by the command "ip a" inside the container:

docker exec -it bind9 ip a

From an external machine, you specify the port from the container creation command and the IP address of your docker host.

If the command is typed without errors, and the server does not respond, then the reason for the error can be searched in the DNS server log with the command

docker logs bind9

Another verification option:

* launch the terminal inside the container with the command docker exec -it bind9 bash
* install nslookup inside the container with the command: apt-get install dnsutils
* and contact the server from inside the container with a simple question

nslookup localhost 127.0.0.1

* and then with a more complex

nslookup ComputerName1.your\_domain\_name 127.0.0.1

* inside the container, you can restart bind with the “service named reload” command and view the status with the command: service named status