Vagrant multi machine assignment

Multi-machine setup

Vagrant can reproduce a production environment by using the "multi machine" feature. Several virtual machines will be created, with just one Vagrantfile. It is highly recommended to use this feature. See the documentation for more details.

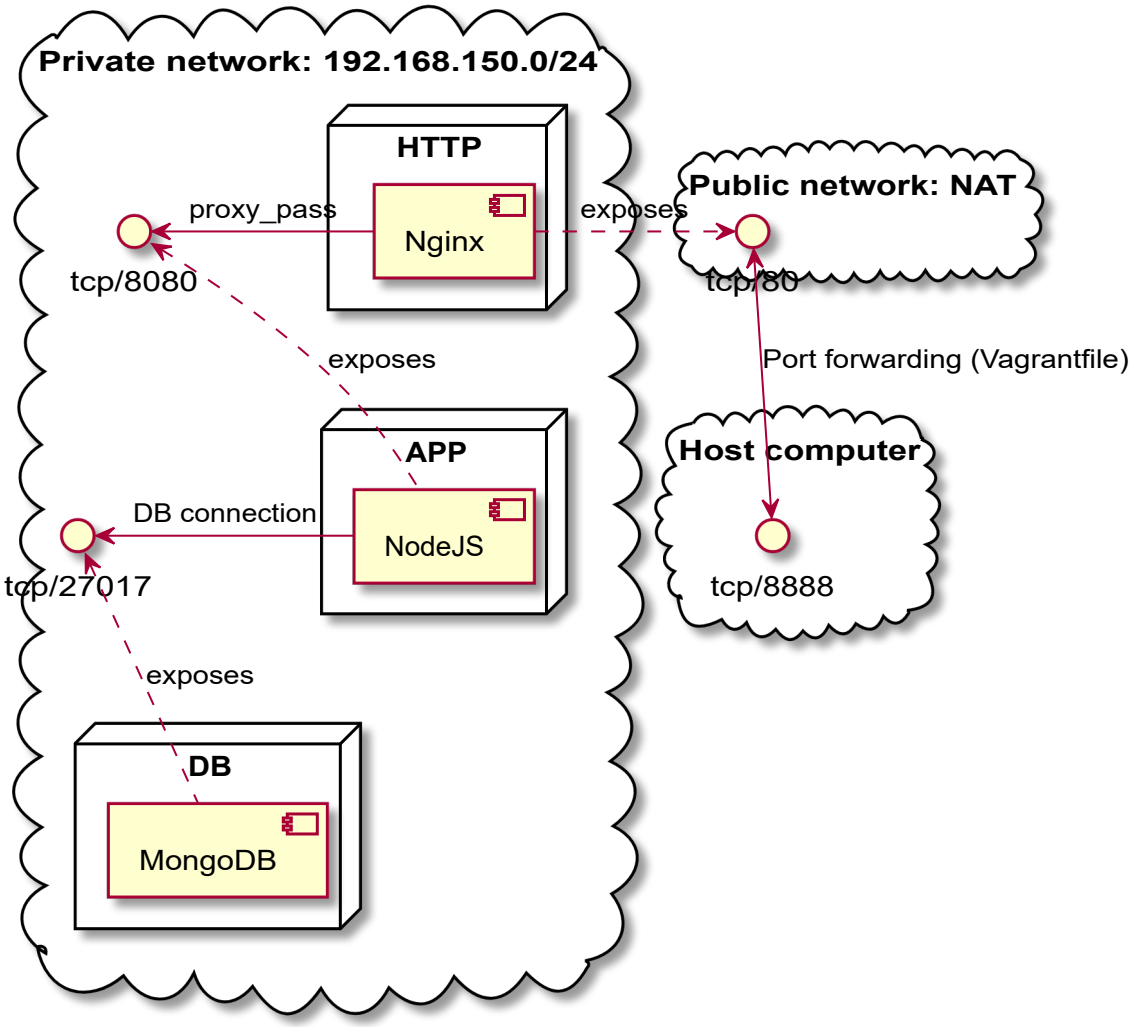
```
config.vm.define "todoapp" do |todoapp|
    todoapp.vm.provider "virtualbox" do |vb|
        vb.name = "TODO_APP_4640"
        vb.memory = 2048
    end
    todoapp.vm.hostname = "todoapp.bcit.local"
    todoapp.vm.network "private_network", ip: "192.168.150.10"
    todoapp.vm.network "forwarded_port", guest: 80, host: 12080
    todoapp.vm.provision "file", source: "./files/nginx.conf", destination:
"/home/admin/nginx.conf"
    # todoapp.vm.provision "shell", path: "./files/vagrant_appsetup.sh"
    # Do other machine-specific provisioning here
end

config.vm.define "tododb" do |tododb|
    tododb.vm.provider "virtualbox" do |vb|
        vb.name = "TODO_DB_4640"
        vb.memory = 1536
    end
    tododb.vm.hostname = "tododb.bcit.local"
    tododb.vm.provision "file", source: "./files/mongorepo.txt", destination:
"/home/admin/mongorepo.txt"
    # tododb.vm.provision "shell", path: "./files/vagrant_dbsetup.sh"
    # Do other machine-specific provisioning here
end
```

This will create two virtual machines: `TODO_APP_4640` and `TODO_DB_4640`. They will have different amounts of memory, and the `TODO_APP_4640` will also be connected to a private network and given the IP address 192.168.150.10 on this network.

Advanced network configuration

Vagrant can also set up different networks to isolate the different components of our application. Below is a typical setup for a three-tier web application.



Assignment

Your objective is to create the `Vagrantfile` that will create 3 machines:

- the reverse proxy (running Nginx)
- the todoapp service (running NodeJS)
- the database service (running Mongo)

The three virtual machines must share a private network. Use the `private_network` option and set an IP address for each service in the Vagrantfile.

You can use the basebox that we created earlier during the Vagrant lab. The database service must be loaded with the provided Mongo export.

Hints

- Make sure all your machines are connected to the same private network.
- Decide on a IP addressing scheme: give your database, application and Nginx machine different IP addresses.
- You will need to adjust the firewall rules on each machine to open the ports for each service. Make sure you only open ports that are necessary!
- You will need to adjust the configuration of Nginx to forward requests to a different virtual machine rather than localhost.

- You will need to adjust the configuration of the todo application to connect to MongoDB running on a different virtual machine.
- Make sure you only provision what is necessary on each virtual machine. It is fine if all your VMs have port 80 open (it comes from the Kickstart file / Packer image).
- MongoDB needs to listen on the network, and not only localhost.

Preload the database

In order to provision the database with initial data, you will need the Mongo export. Unpack it with tar (`tar zxf filename.tgz`) and use `mongorestore` to import the extracted folder into your MongoDB instance:

- `mongorestore -d <database_name> <dump_folder>` (ACIT4640 is the name of the database configured in the database.js file)
- `<database_name>` is the name of the database used by the application (see the `database.js` file)
- `<dump_folder>` is the path to the archive folder with the MongoDB export (that you downloaded above)
- `mongorestore` requires a locale to be set. You can use the environment variable `LANG`: `export LANG=C`.
- If necessary, you will need to install the package that provides the "mongorestore" binary. You can find it with: `dnf whatprovides mongorestore`

Submission guidelines

- Create a new branch `vagrant_lab` in your GitHub repository.
- Push your files to this branch.
- Submit the link to your repo/branch on D2L (text submission).

Suggested folder structure:

```
.                       # ROOT OF YOUR GIT REPOSITORY
├── Vagrantfile
└── files
    ├── acit_admin_id_rsa
    ├── mongorepo.txt
    ├── nginx.conf
    └── todoapp.service
```

The only requirement is to have a `Vagrantfile`. You can setup your supporting files and folders as you see fit.