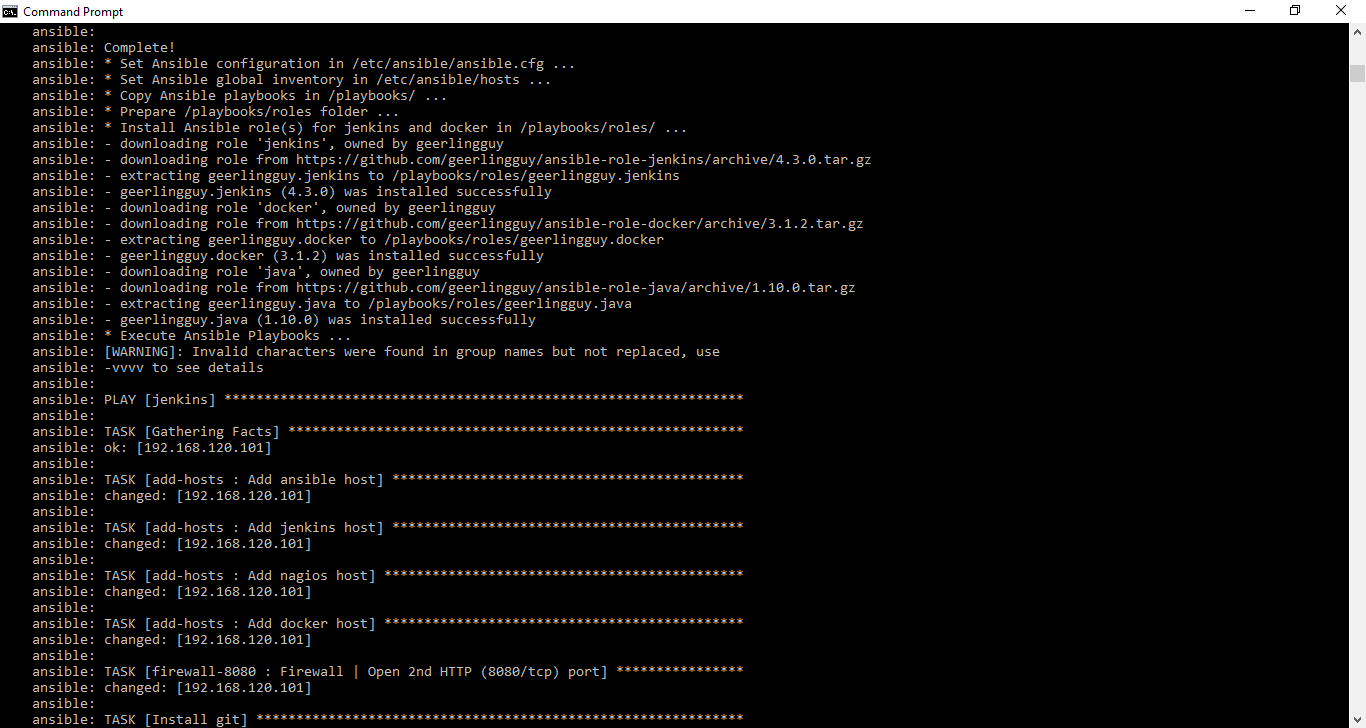
За изпълнение на изпитното задание реших да използвам следния подход: с помощта на Vagrant ще вдигна 4 машини за нуждите на заданието, съответно за всеки инструмент по една машина, а именно Ansible, Docker, Jenkins и Nagios. Машините ще се намират както е указано в една мрежа – 192.168.120.0/24 според заданието и ще са именувани според конвенцията във заданието. Идеята е да си вдигна с помощта на Vagrant всички машини и да провизирам Ansible хоста, който от своя страна да провизира и инсталира останалите необходими хостове, като стартира автоматично набор от playbooks. Като финалното задание за билдването на проект в Jenkins и постигането на крайната цел смятам да изпълня ръчно във фронтенда на Jenkins, както и настойките на Jenkins по отношение на креденшълите и слейв хоста.

По отношение на първия таск прилагам Vagrantfile от който е видно, че си създаваме следните хостове: avm.dob.exam, jvm.dob.exam, dvm.dob.exam и nvm.dob.exam. Както и добаването на мрежа, пренасочването на портове и добавянето на синхронизирана папка за обмен на файлове. Само за Ansible хоста нещата стоят по различно там си инсталирам самия инструмент и си го настройвам, както и добавям набор от файлове за провизирането на останaлите машини и накрая стартирам процеса по провизиране.

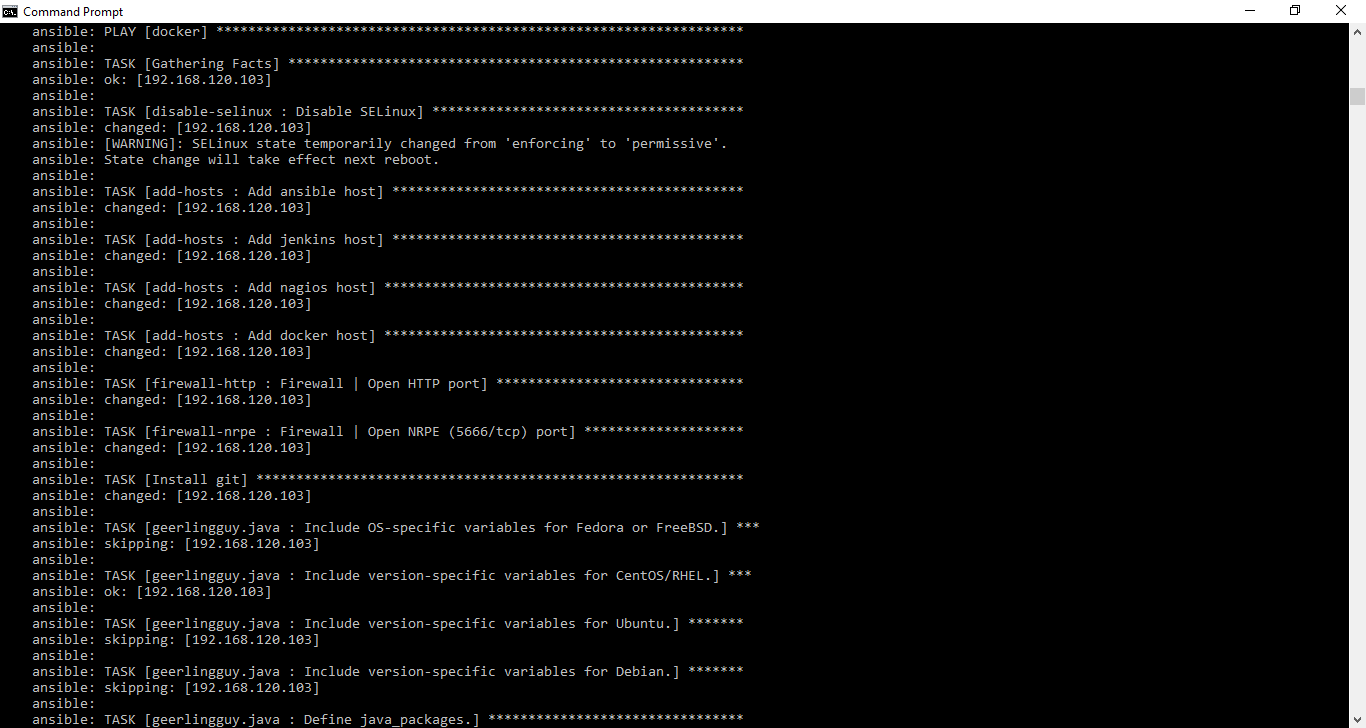
# -\*- mode: ruby -\*-  
# vi: set ft=ruby :  
  
Vagrant.configure("2") do |config|  
  
 config.ssh.insert\_key = *false* config.vm.define "jenkins" do |jenkins|  
 jenkins.vm.box="shekeriev/centos-8-minimal"  
 jenkins.vm.hostname = "jvm.dob.exam"  
 jenkins.vm.network "private\_network", *ip:* "192.168.120.101"  
 jenkins.vm.network "forwarded\_port", *guest: 8080*, *host: 8000*, *auto\_correct: true* jenkins.vm.synced\_folder "vagrant/jenkins/", "/vagrant"  
 end  
   
 config.vm.define "nagios" do |nagios|  
 nagios.vm.box="shekeriev/centos-8-minimal"  
 nagios.vm.hostname = "nvm.dob.exam"  
 nagios.vm.network "private\_network", *ip:* "192.168.120.102"  
 nagios.vm.network "forwarded\_port", *guest: 80*, *host: 8001*, *auto\_correct: true* nagios.vm.synced\_folder "vagrant/nagios/", "/vagrant"  
 end  
   
 config.vm.define "docker" do |docker|  
 docker.vm.box="shekeriev/centos-8-minimal"  
 docker.vm.hostname = "dvm.dob.exam"  
 docker.vm.network "private\_network", *ip:* "192.168.120.103"  
 docker.vm.network "forwarded\_port", *guest: 80*, *host: 8002*, *auto\_correct: true* docker.vm.synced\_folder "vagrant/docker/", "/vagrant"  
 end  
  
 config.vm.define "ansible" do |ansible|  
 ansible.vm.box="shekeriev/centos-8-minimal"  
 ansible.vm.hostname = "avm.dob.exam"  
 ansible.vm.network "private\_network", *ip:* "192.168.120.100"  
 ansible.vm.synced\_folder "vagrant/ansible/", "/vagrant"  
 ansible.vm.provision "shell", *inline:* <<EOS  
echo "\* Add hosts ..."  
echo "192.168.120.100 avm.dob.exam ansible" >> /etc/hosts  
echo "192.168.120.101 jvm.dob.exam jenkins" >> /etc/hosts  
echo "192.168.120.102 nvm.dob.exam nagios" >> /etc/hosts  
echo "192.168.120.103 dvm.dob.exam docker" >> /etc/hosts  
  
echo "\* Install Ansible ..."  
dnf install -y epel-release  
dnf install -y ansible  
  
echo "\* Set Ansible configuration in /etc/ansible/ansible.cfg ..."  
cp /vagrant/ansible.cfg /etc/ansible/ansible.cfg  
  
echo "\* Set Ansible global inventory in /etc/ansible/hosts ..."  
cp /vagrant/hosts /etc/ansible/hosts  
  
echo "\* Copy Ansible playbooks in /playbooks/ ..."  
cp -R /vagrant/playbooks /playbooks  
  
echo "\* Prepare /playbooks/roles folder ..."  
cp -R /vagrant/roles /playbooks  
  
echo "\* Install Ansible role(s) for jenkins and docker in /playbooks/roles/ ..."  
ansible-galaxy install geerlingguy.jenkins -p /playbooks/roles/  
ansible-galaxy install geerlingguy.docker -p /playbooks/roles/  
ansible-galaxy install geerlingguy.java -p /playbooks/roles/  
  
echo "\* Execute Ansible Playbooks ..."  
ansible-playbook /playbooks/install-all.yml  
  
EOS  
 end  
end

Картинки относно изпълнението на Vagrantfile ще е лудост да добавям, тъй като екраните са много пък и за голяма част от нещата съм използвам шаблони от подготовката за изпита, тъй като нещата ми се сториха най-изчистени.

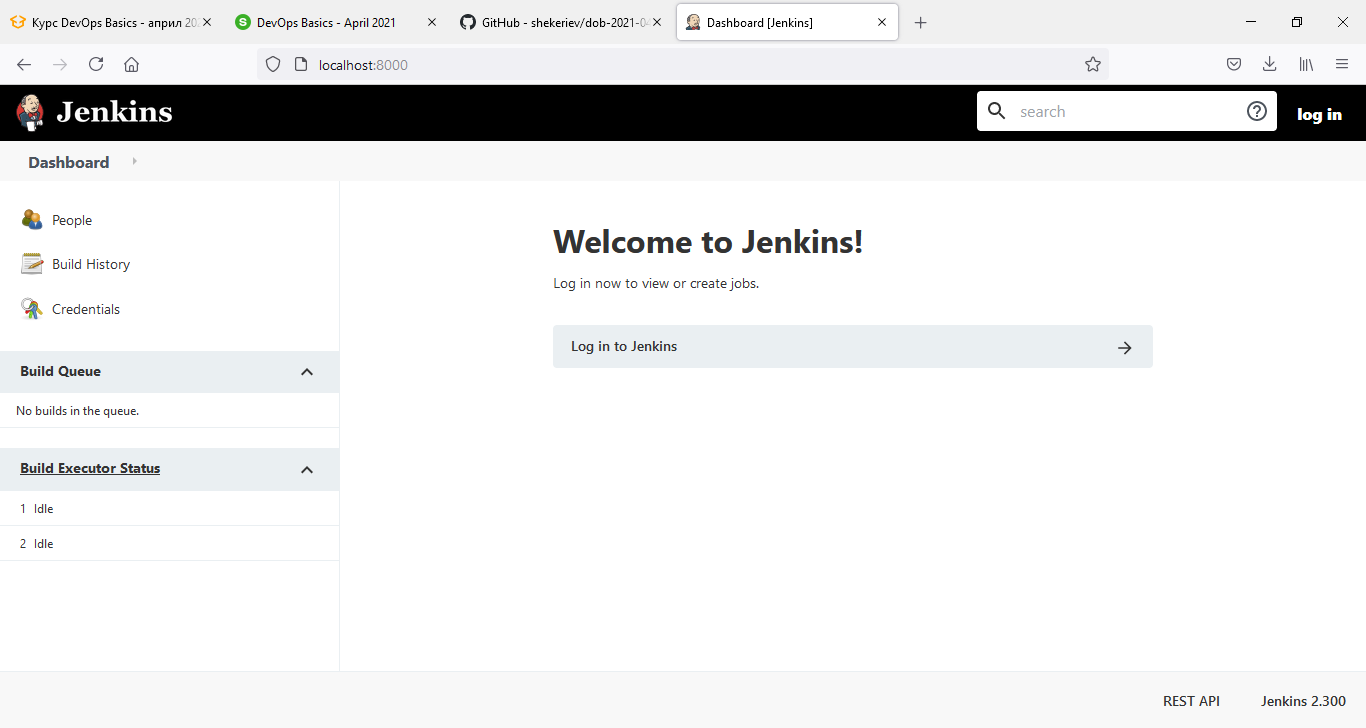
По отношение на работенето и провизирането от страна на Ansible ще добавя няколко картинки за да се види, че работи. По детайлна информация има във набора от файлове приложени към решението, набор от плейбуци и роли, както и необходими файлове за настройка на инструментите.

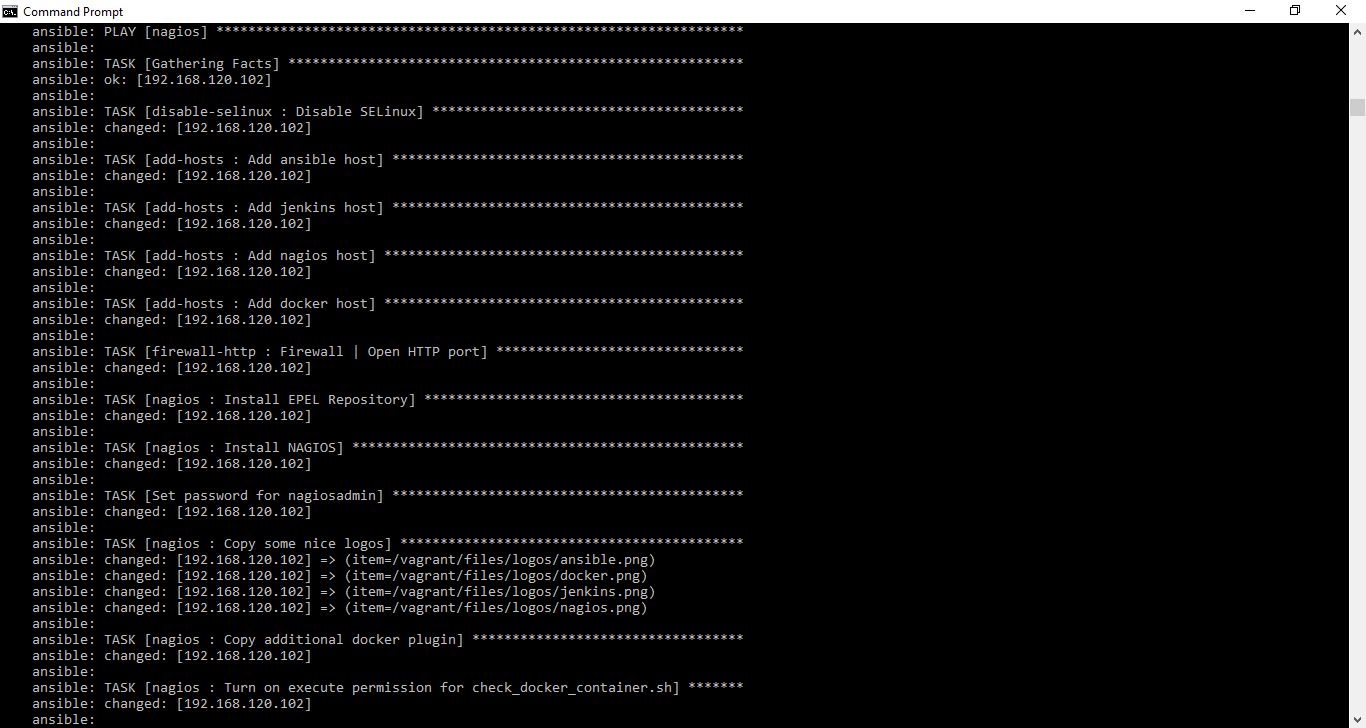


От горния екран е видно че Ansible е готов зарежда чуджите роли от Galaxy и започва инсталирането и провизирането на Jenkins.

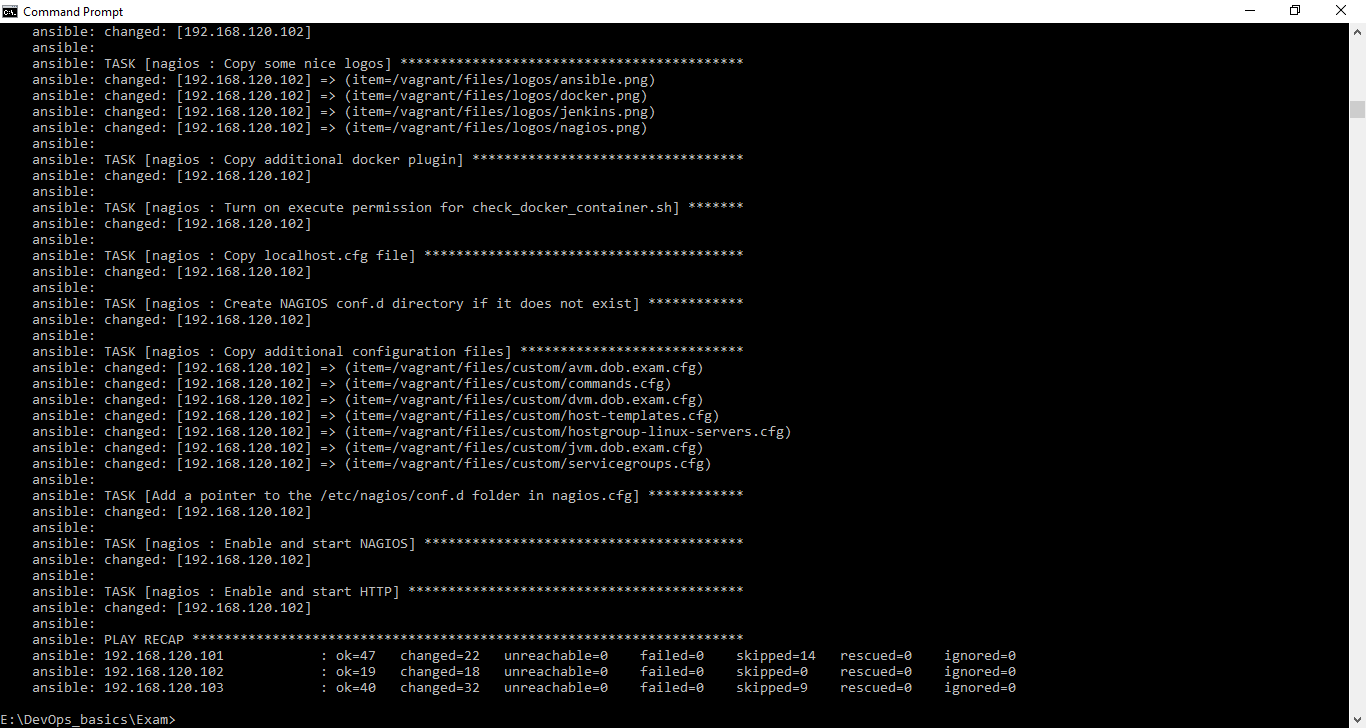


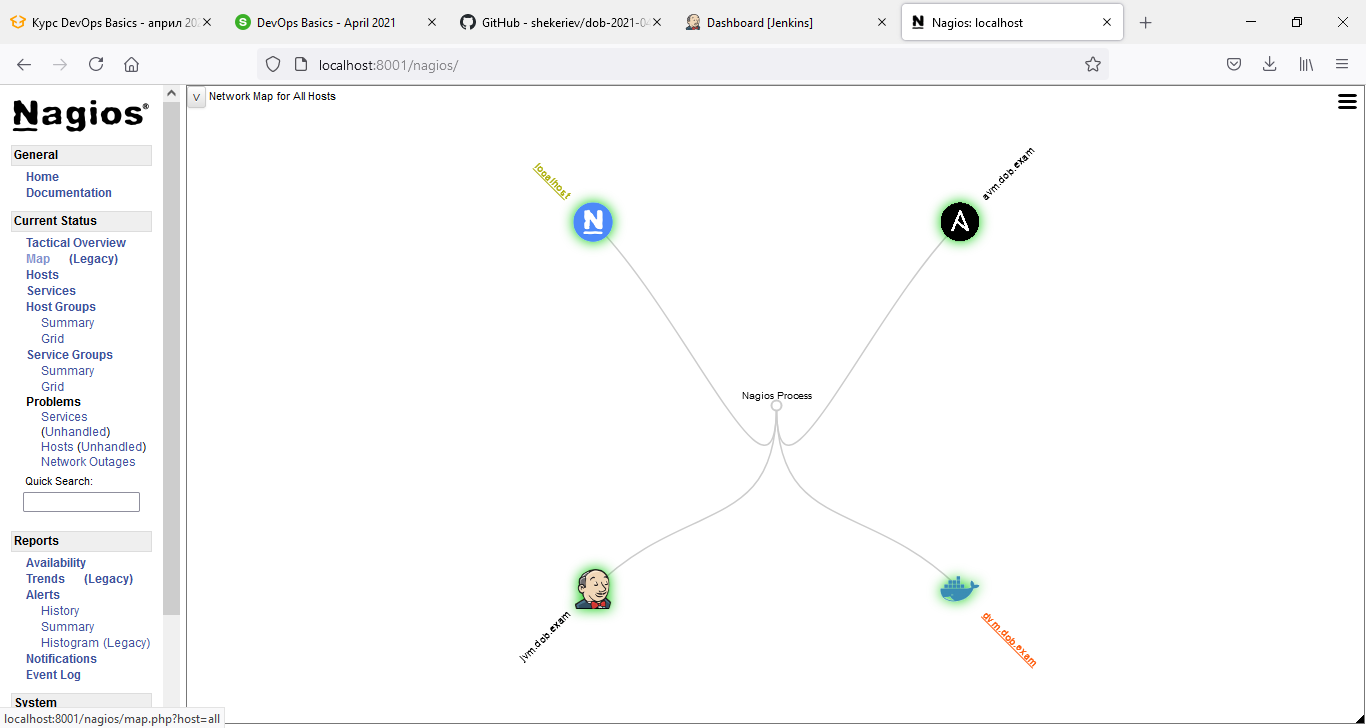
От горния екран е видно че започва и инсталирането и провизирането на Docker , което означaва, че имам вече готов Jenkins и мога да го достъпя на адрес - <http://localhost:8000>

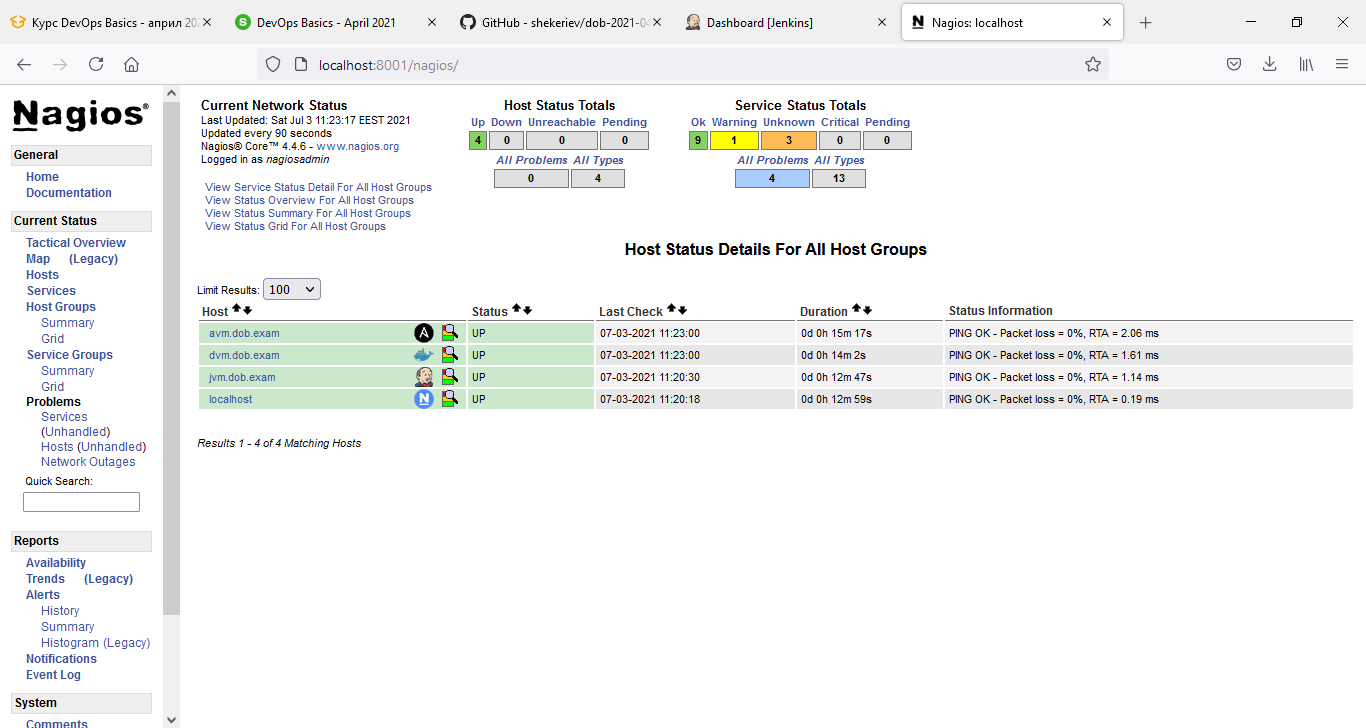


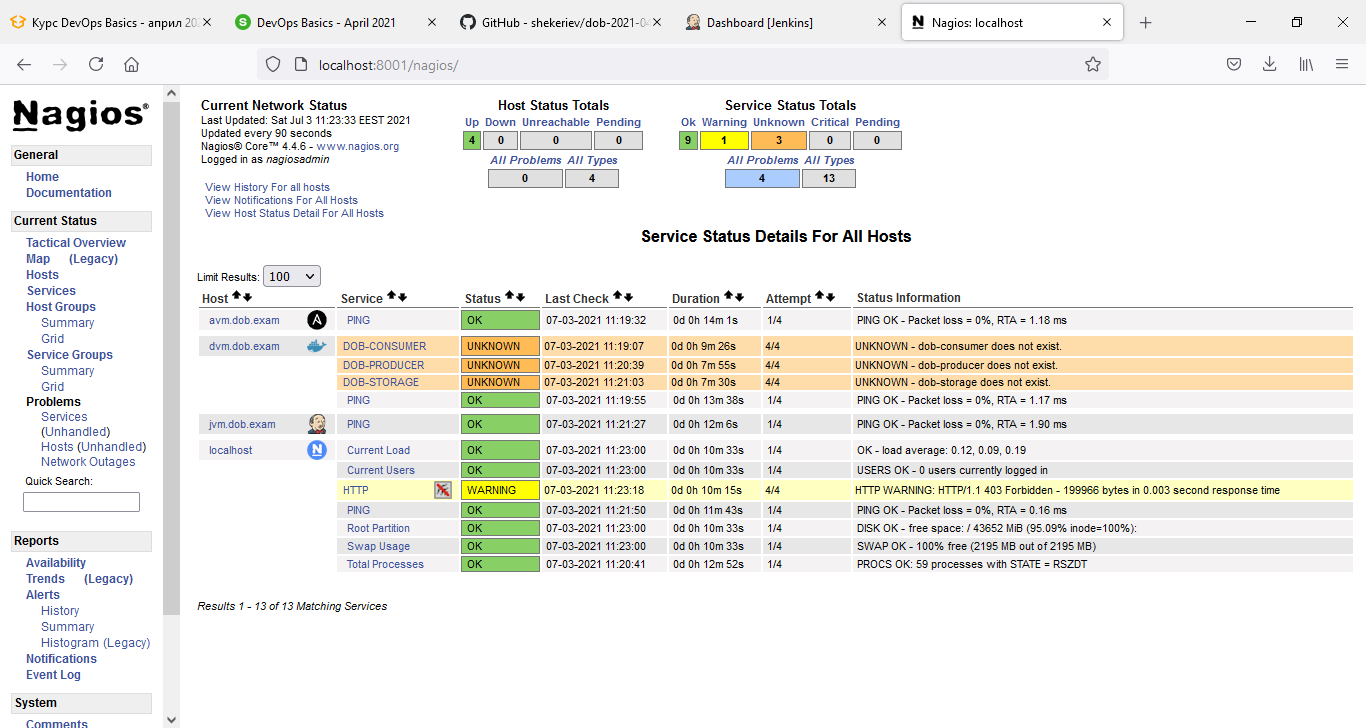


От последния екран е видно, че започва инсталирането и провизирането на Nagios, което означава че и Docker е готов.

 и тук вече се вижда че и Nagios е готов и на адрес - <http://localhost:8001/nagios> може да го достъпя. Необходимите креденшъли са Username: nagiosadmin , Password: admin.







С полседните екрани се вижда, че и Nagios работи добавени са хостовете и на картат и се вижда че наблюдението по PING вече работи, не работ само наблюдението на контейнерите тъй като ги няма самите контейнери. Това вече смятам да направя ръчно през Jenkins.

Сега първо влизаме в Jenkins със следните креденшъли: Username: admin, password: admin те са създадени автоматично от чуждата роля, която използвам за да си инсталирам и провизирам Jenkins а именно - geerlingguy.jenkins. Тази и други две роли се взети от Ansible Galaxy от репото на Jeff Geerling.

След като вече съм вътре отивам първо да си добавя Global Credentials:

Отивам на **Manage Jenkins** след това **Manage Credentials** избирам **Jenkins**  и след това **Global credentials (unrestricted)** след това **Add Credentials** и добавям следните креденшъли:

* Kind: **Username with password**
* Scope: **Global**
* Username: **vagrant**
* Password: **vagrant**

Потвърждавам с **OK**

След това трябва да добавя хост:

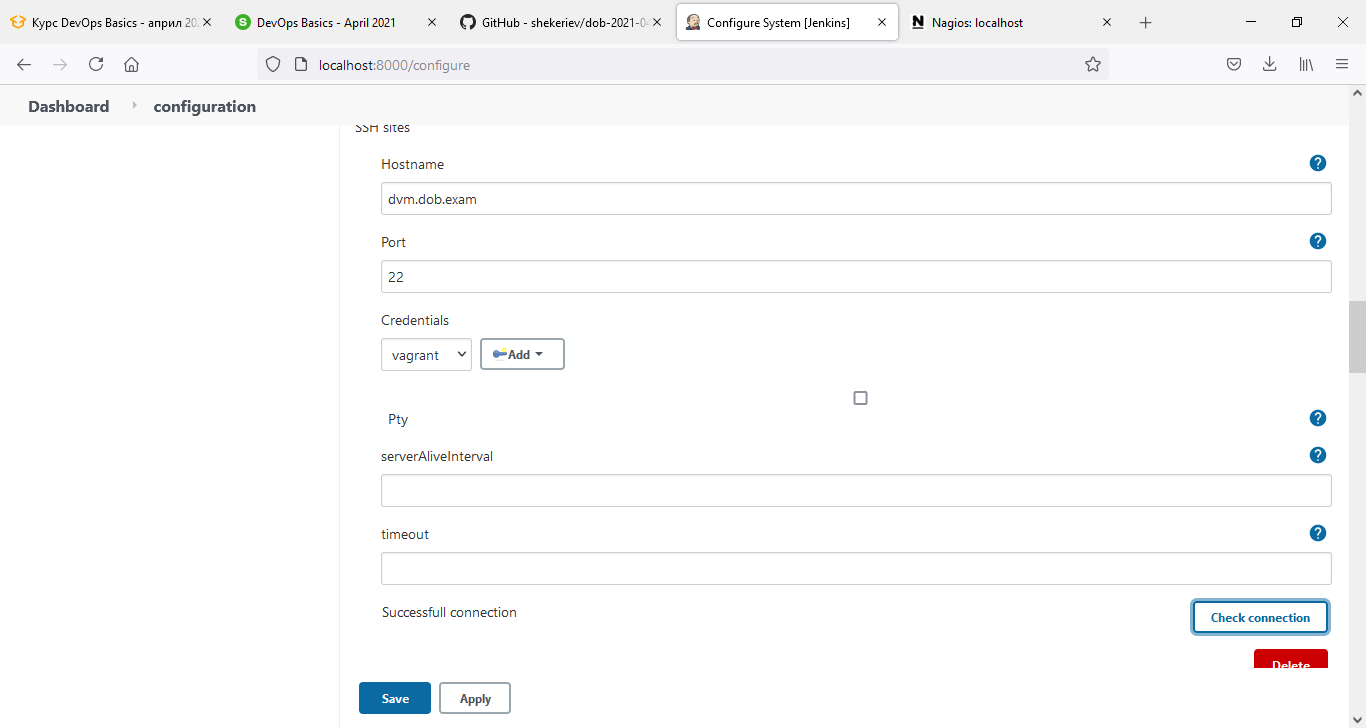
Отивам **Manage Jenkins** след това **Configure System**

Отивам **SSH remote hosts** секция и с **Add** бутона добавям

* Hostname: **dvm.dob.exam**
* Port: **22**
* Credentials: **vagrant**

Тествам връзката с **Check connection**

Потвърждавам с **Save** бутона



След това трябва да добавя този хост като Slave в Jenkins.

Избирам **Manage Jenkins** след това **Manage Nodes and Clouds**

Избирам **New Node**

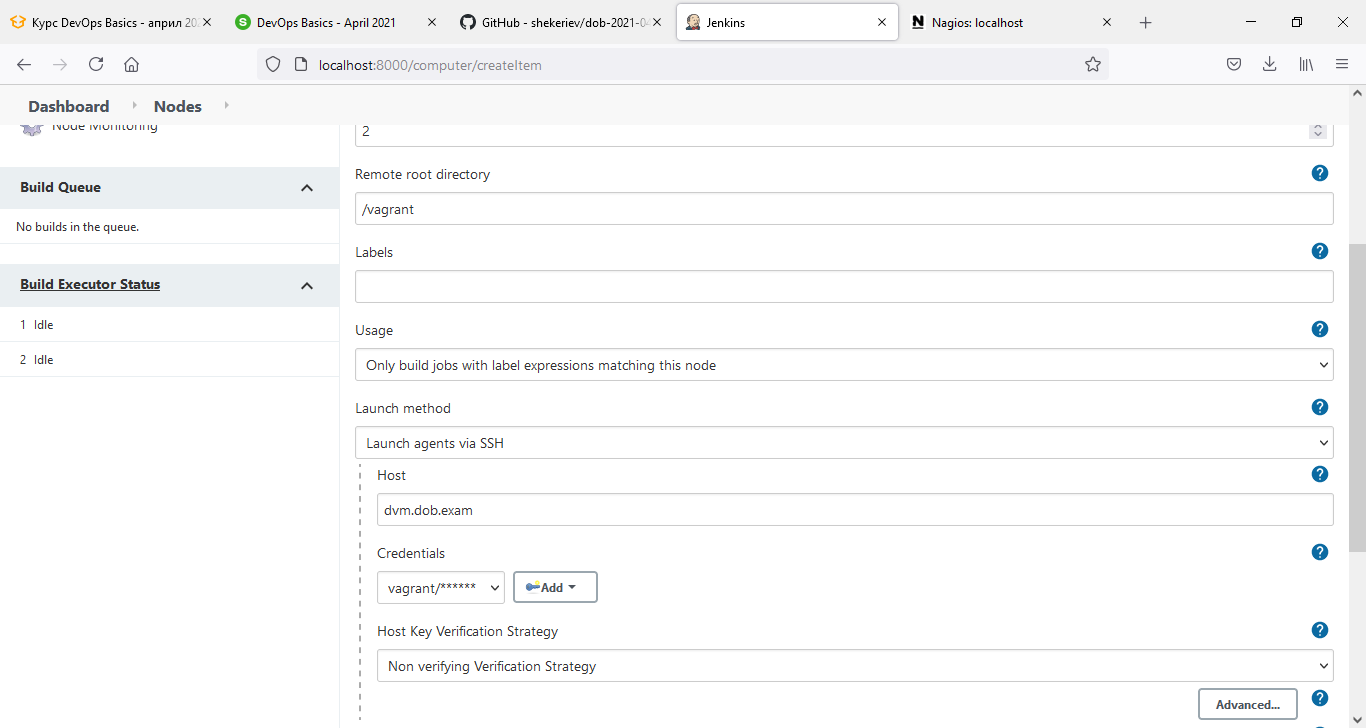
Добавям **docker** в **Node name** и избирам **Permanent Agent**

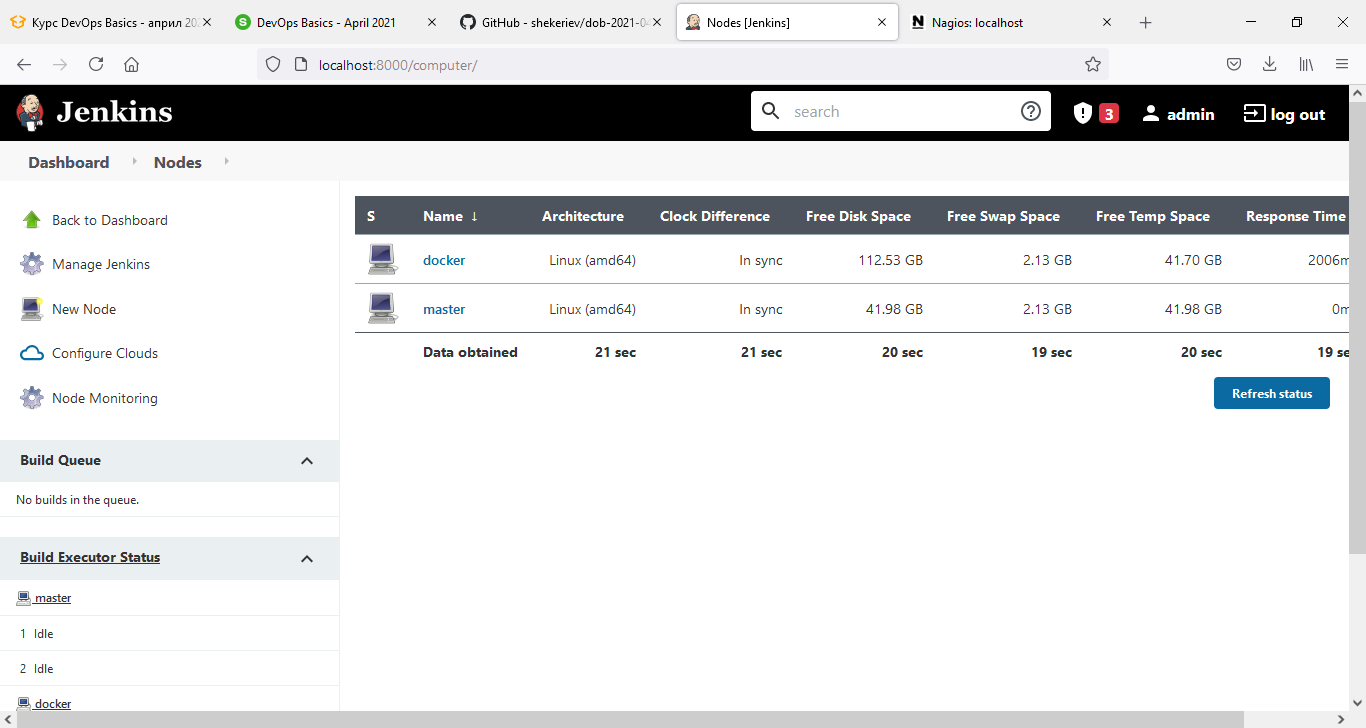
Потвърждавам с **OK**

Въвеждам следните настройки:

* # of executors: **2**
* Remote root directory: **/vagrant**
* Labels: **docker**
* Usage: **Only build jobs with label expressions matching this node**
* Launch method: **Launch agents via SSH**
* Host: **dvm.dob.exam**
* Credentials: **vagrant**
* Host Key Verification Strategy: **Non verifying Verification Strategy**

Потвърждаваме със **Save**





С което Jenkins вече е нагласен и може да се залавям с главната цел да си вдигна автоматично приложението от Github чрез Jenkins Project.

Избирам **New Item**

Втвеждам следните неща:

* Name: DOB-EXAM-20210703
* Type: **Freestyle Project**

Потвърждавам с **OK**

