Today I am completing my studies at the SUSE Cloud Native Foundations Scholarship Program

<https://www.udacity.com/scholarships/suse-cloud-native-foundations-scholarship>

and they teach CI / CD, Kubernetes, dockers, both in theory and in practice, and so I recorded some learning outcome of a beginner in the form of a video review of the final demo result, he is now able to deploy complex infrastructure system solutions in minutes.

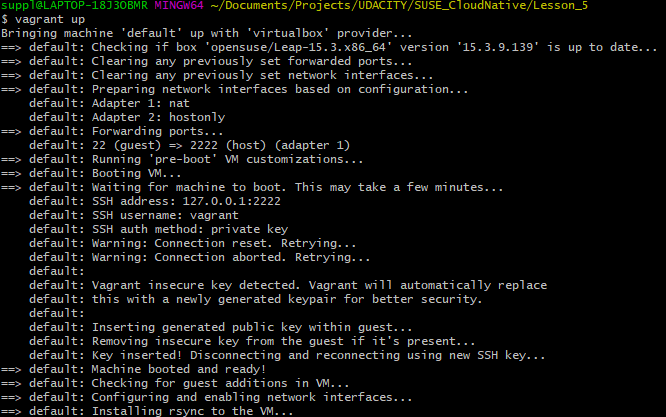
<https://youtu.be/aIoQmAy9k4Q>

And some step by step description on some exercises from the final lesson5 in the udacity online:

Finally, I reached my desktop computer and did the last lesson exercises, installed my computer's BIOS virtualization options. also, I don't have so much free RAM memory(and only 2 CPU cores), so I made some modifications to the Vagrant settings file

<https://github.com/ryuriymega/python-helloworld/blob/main/Vagrantfile>

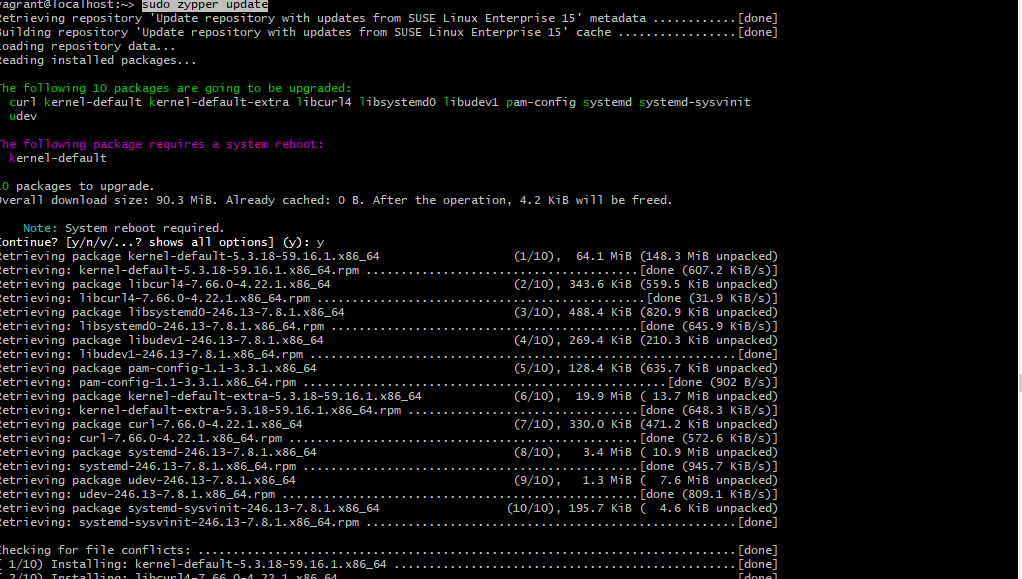
like in the github file from the above url(also I used the latest OS of SUSE which I found on the Vagrant site [Vagrant Cloud by HashiCorp](https://app.vagrantup.com/opensuse/boxes/Leap-15.3.x86_64) ) and then I used [ vagrant up ] to begin.



then I used [ vagrant ssh ] to get into console of the installed virtual machine

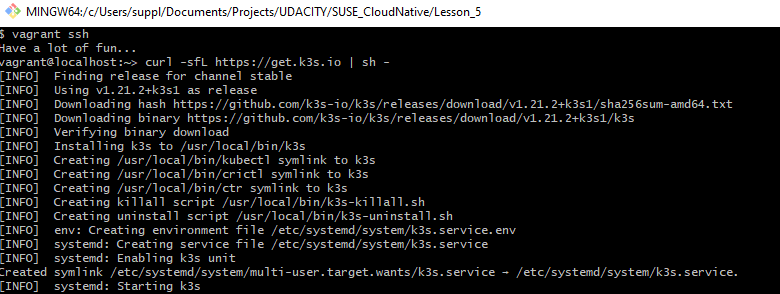
need to update the OS of SuSe in the virtual machine via vagrant ssh like this:

[ sudo zypper update ]



and to install the k3s like in the instructions on the site [Lightweight Kubernetes](https://k3s.io/)

[ chmod oug+r /etc/rancher/k3s/k3s.yaml ]



[ k3s kubectl get node ]

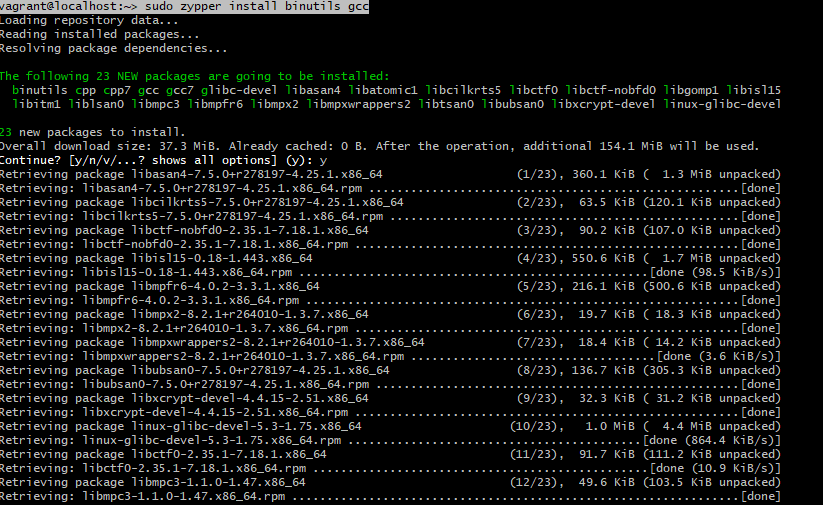


now we are ready to prepare before installing the ArgoCd [Getting Started - Argo CD - Declarative GitOps CD for Kubernetes](https://argoproj.github.io/argo-cd/getting_started/)

................................................[binutils and gcc]................................................

to resolve future dependencies need to install gcc and bin with a command like below:

[ sudo zypper install binutils gcc git docker openssl ]



................................................[ brew ]................................................

then we will need a [ brew ] , commands in the terminal to complete it:

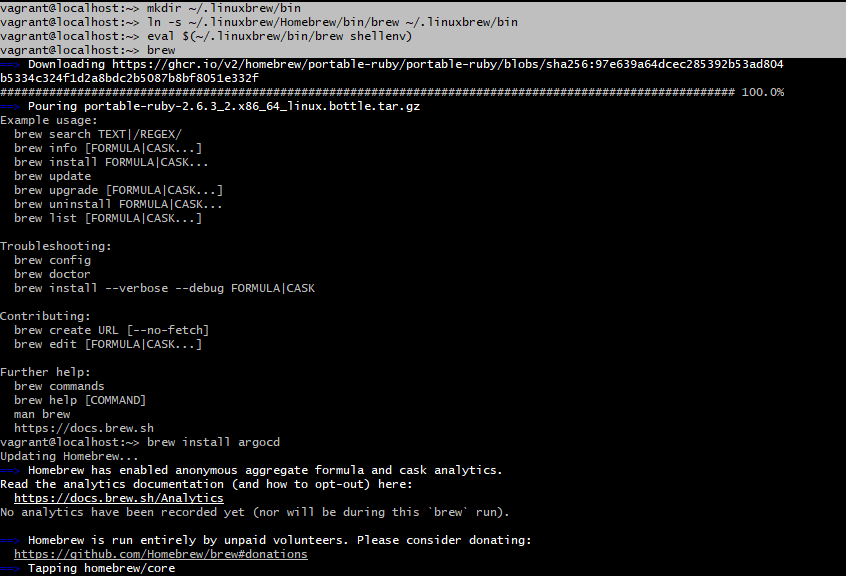
[ git clone [GitHub - Homebrew/brew: The missing package manager for macOS (or Linux)](https://github.com/Homebrew/brew) ~/.linuxbrew/Homebrew ]

[ mkdir ~/.linuxbrew/bin ]

[ ln -s ~/.linuxbrew/Homebrew/bin/brew ~/.linuxbrew/bin ]

[ eval $(~/.linuxbrew/bin/brew shellenv) ]

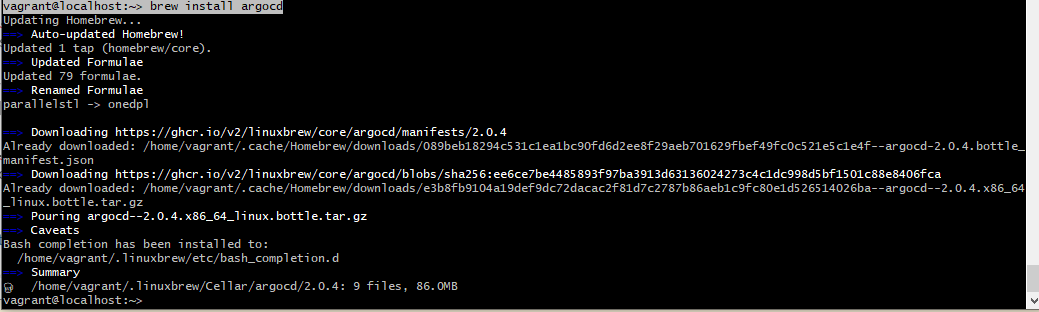
[ brew ]



................................................[ ArgoCd]................................................

when brew was installed then we ready to install the argocd:

[ brew install argocd ]



[ kubectl create namespace argocd ]



[ kubectl apply -n argocd -f <https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml> ]

[ sudo reboot ]

then get password for login in argocd

[ kubectl -n argocd get secret argocd-initial-admin-secret -o jsonpath="{.data.password}" | base64 -d ]

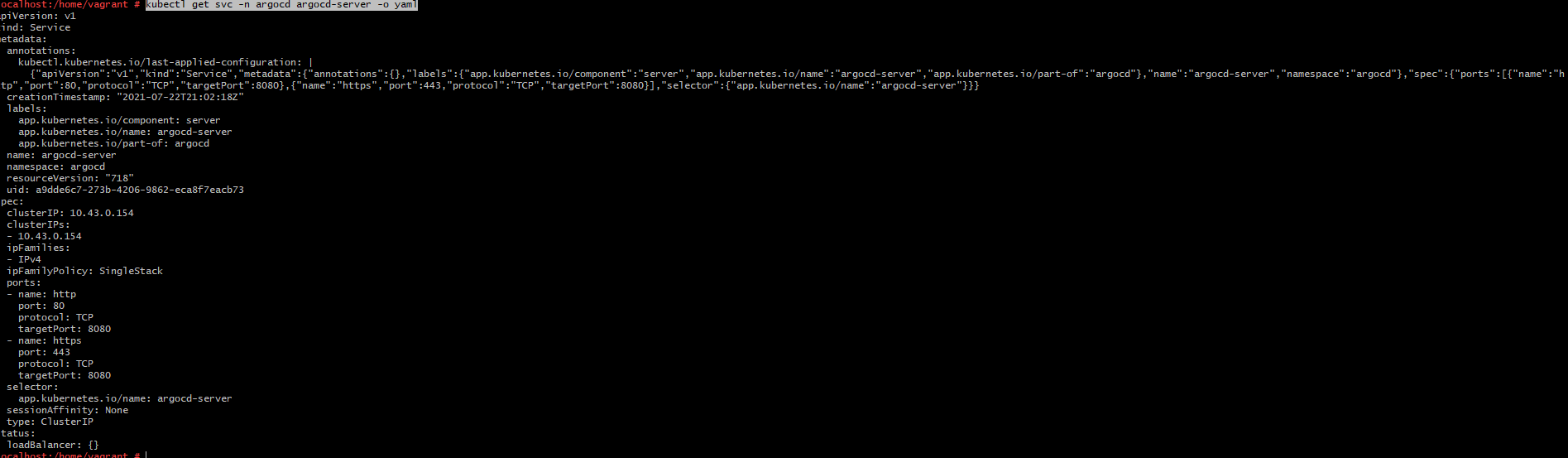
you can see my password is :

uUwRlmgwJKnoyVyN



check

[ kubectl get svc -n argocd argocd-server -o yaml ]



yaml

[ kubectl get svc -n argocd argocd-server -o yaml > argocd-nodepprt.yml ]

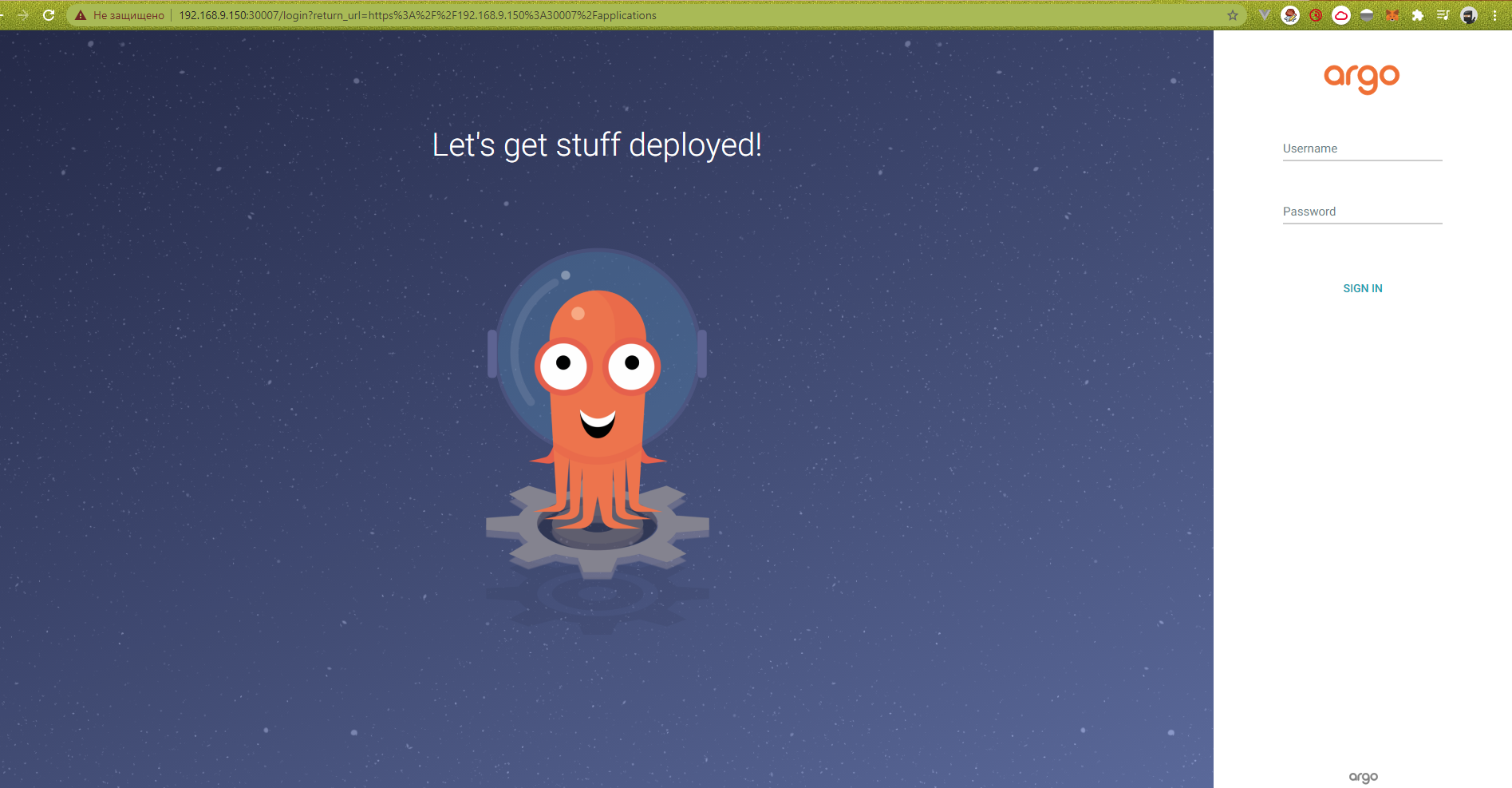
edit recently generated yaml manifest with type of NodePort ( check docs.: [Service](https://kubernetes.io/docs/concepts/services-networking/service/) )

[ argocd-nodepprt.yml ]

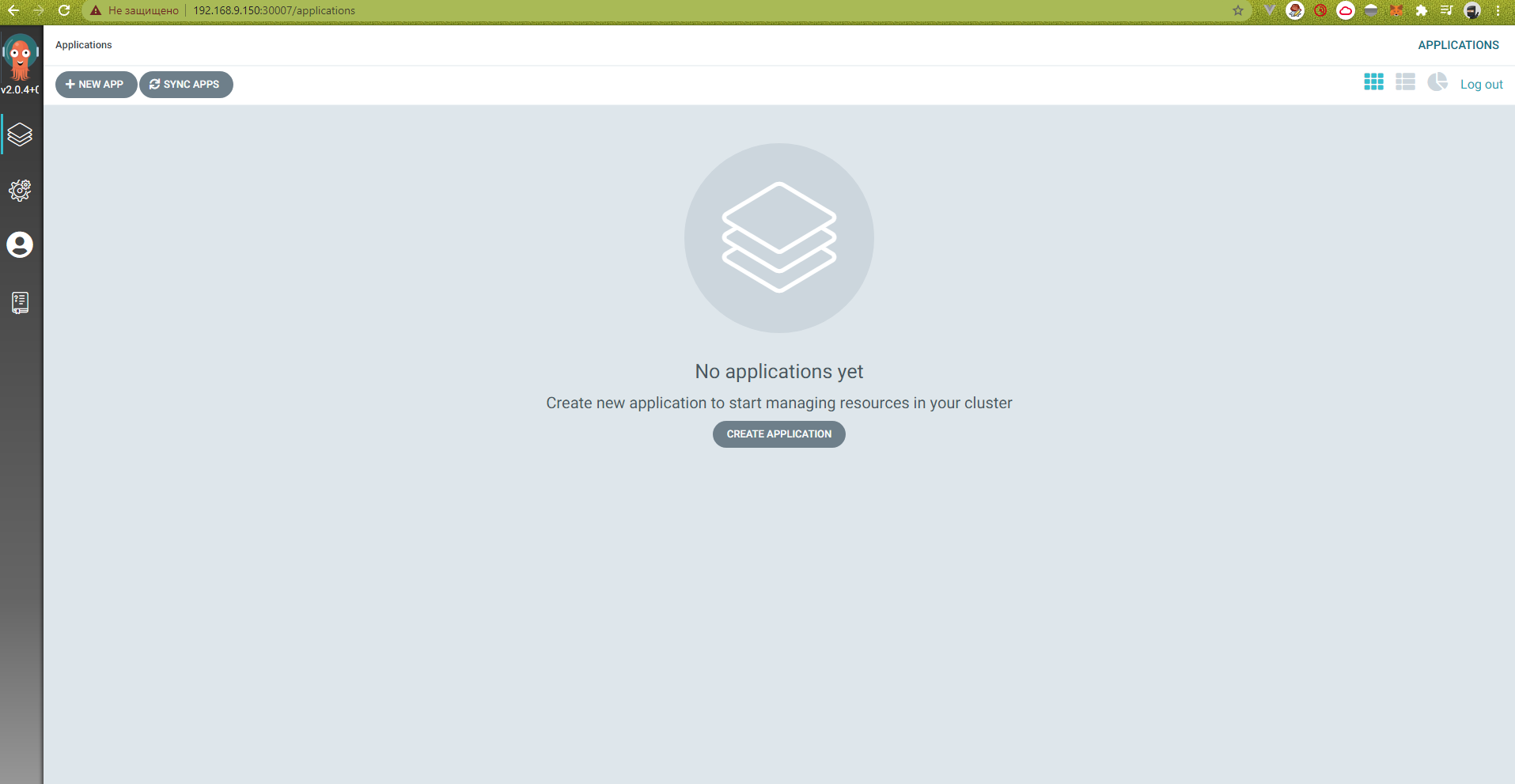
apply

[ kubectl apply -f argocd-nodepprt.yml ]

finally the argocd ready to use on port of 30007



used recently recieved password to login with admin account:



manifest for test deploy [python-helloworld/deployment.yaml at main · ryuriymega/python-helloworld](https://github.com/ryuriymega/python-helloworld/blob/main/manifest/deployment.yaml)

[ vim argocd-python.yaml ]

kind: Application

metadata:

name: python-helloworld

namespace: argocd

spec:

destination:

namespace: default

server: [https://kubernetes.default.svc](https://kubernetes.default.svc/)

project: default

source:

path: manifest

repoURL: [GitHub - ryuriymega/python-helloworld: test](https://github.com/ryuriymega/python-helloworld)

targetRevision: HEAD

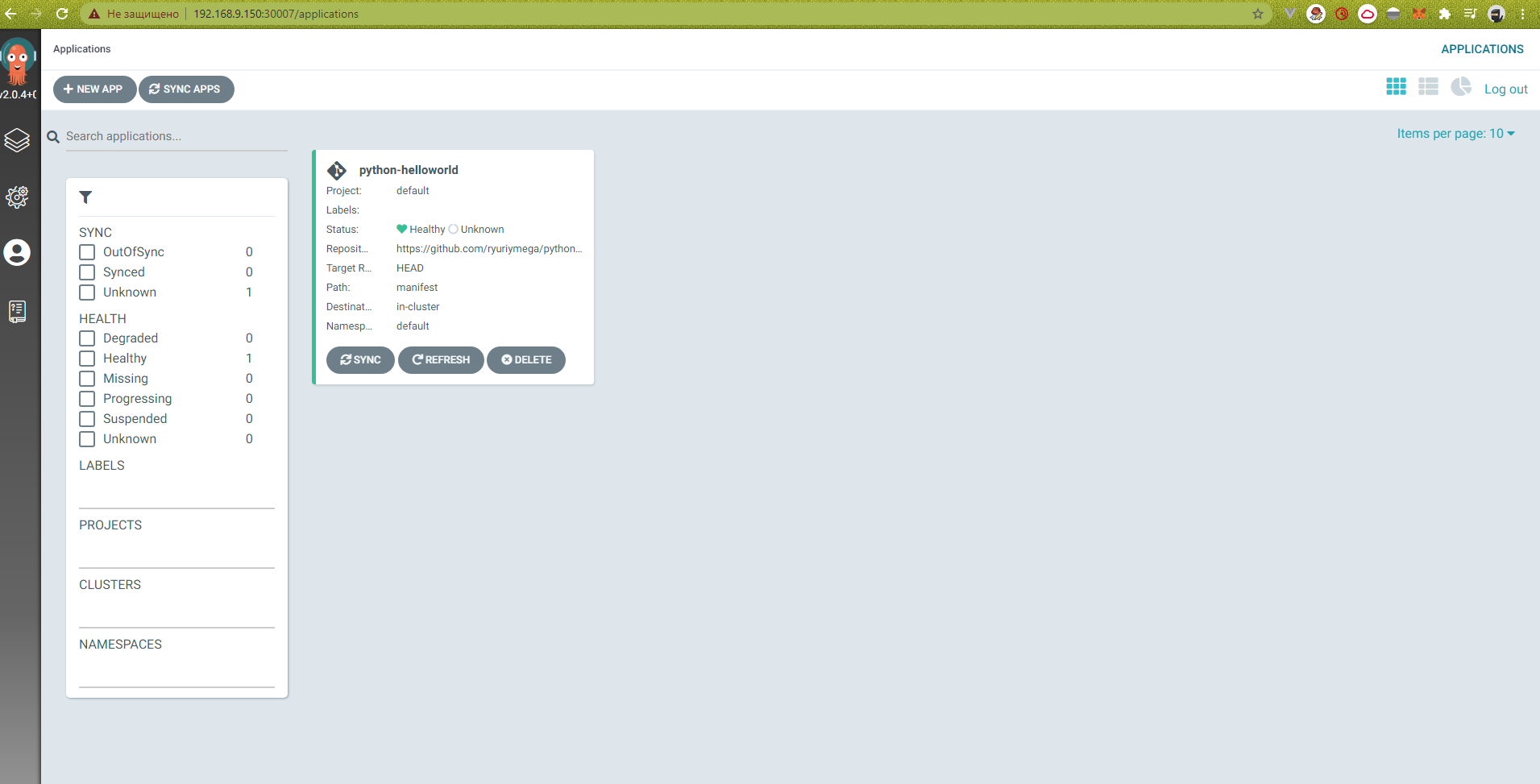
syncPolicy: {}

[ kubectl apply -f argocd-python.yaml ]

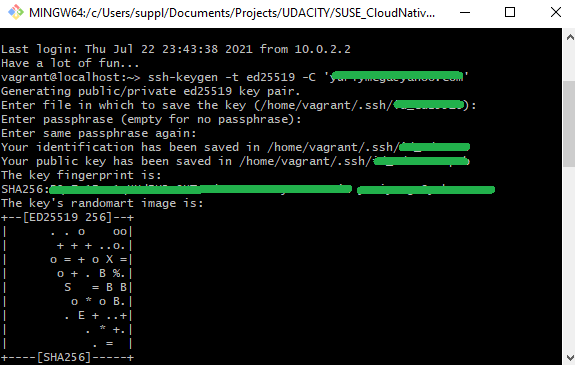
check created app. :

[ kubectl get application -n argocd ]

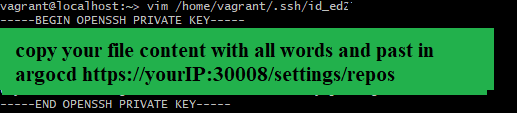




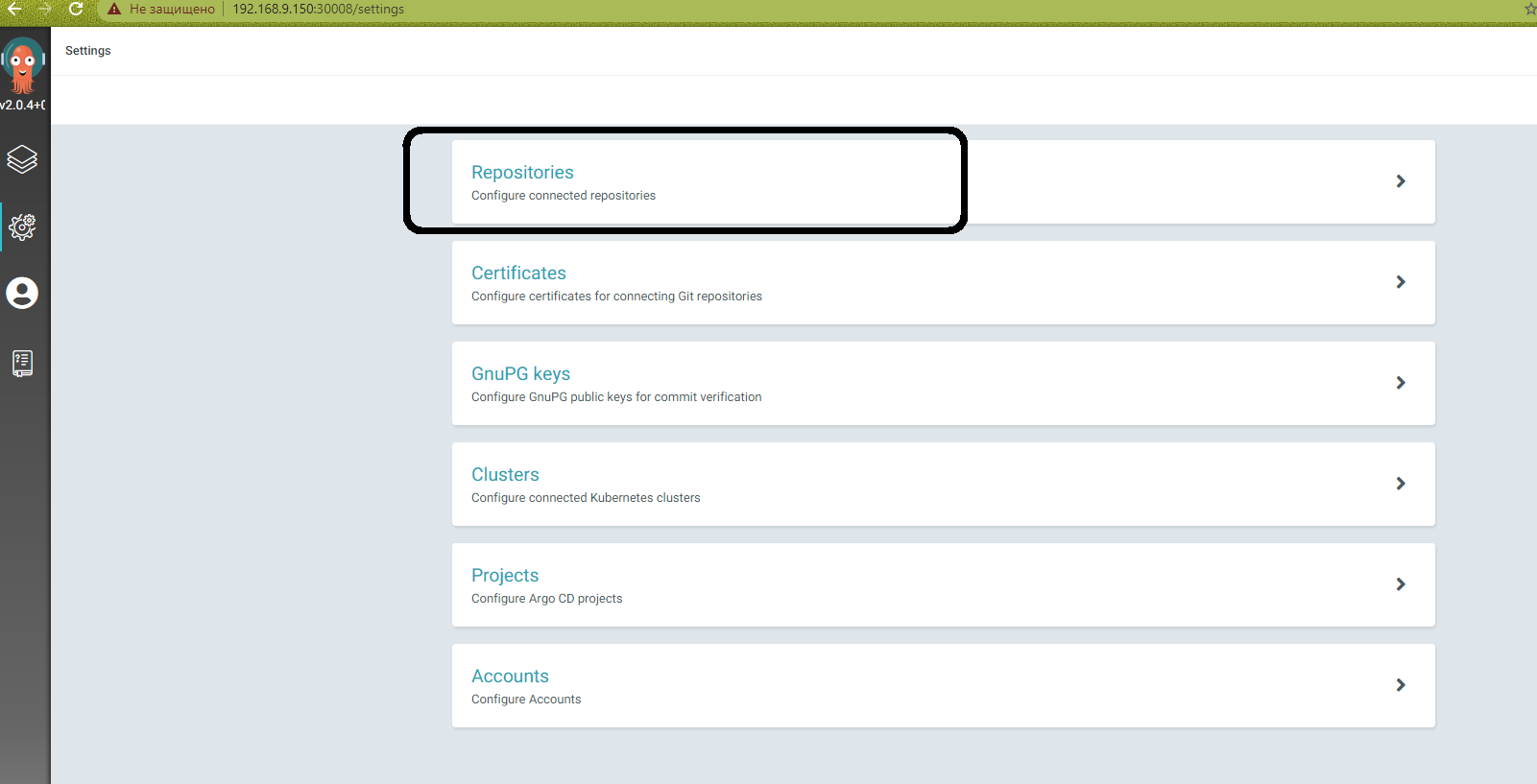
there was an error when I started the argocd with applications/python-helloworld because my github has a secure configuration.  
I generated a key with email which you use in your github:  
[ ssh-keygen -t ed25519 -C '[youremail@google.com](mailto:youremail@google.com)' ]



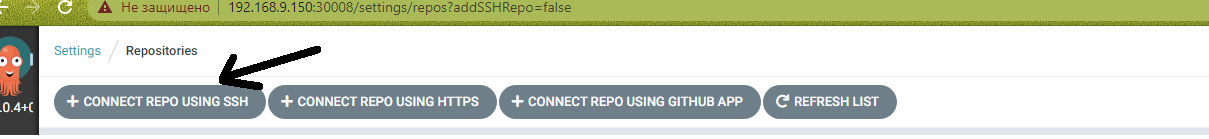
then copy private key  
[ vim /home/vagrant/.ssh/id\_yourGeneratefileIDhere ]



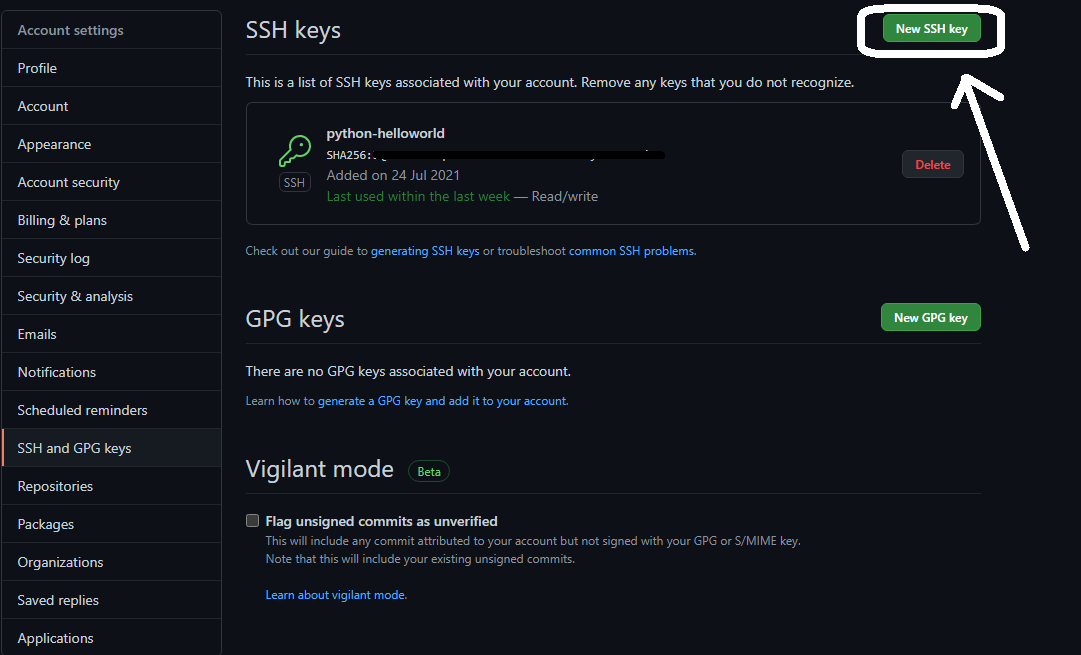
then open your argocd settings and select a Repositories



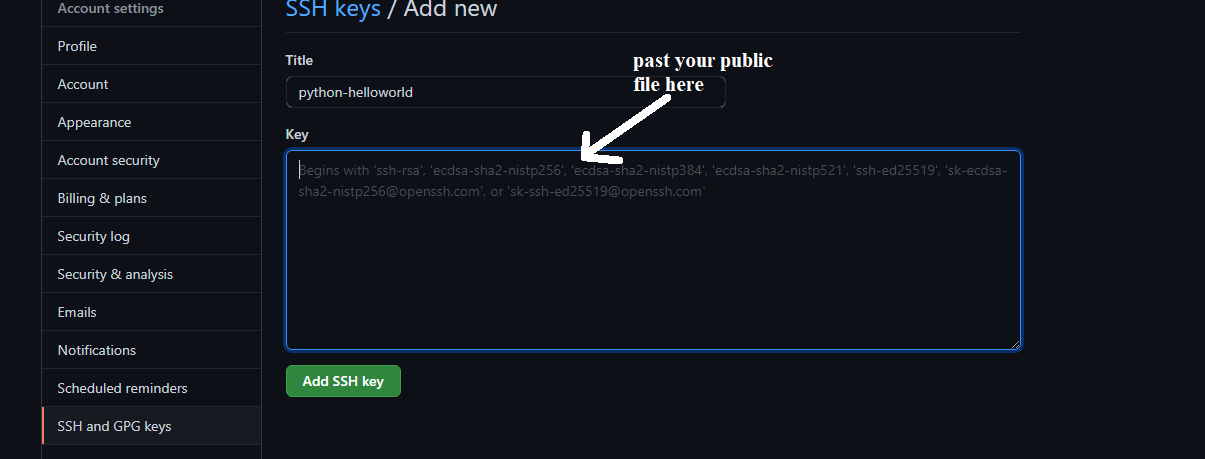
then press { Connect repo using SSH }



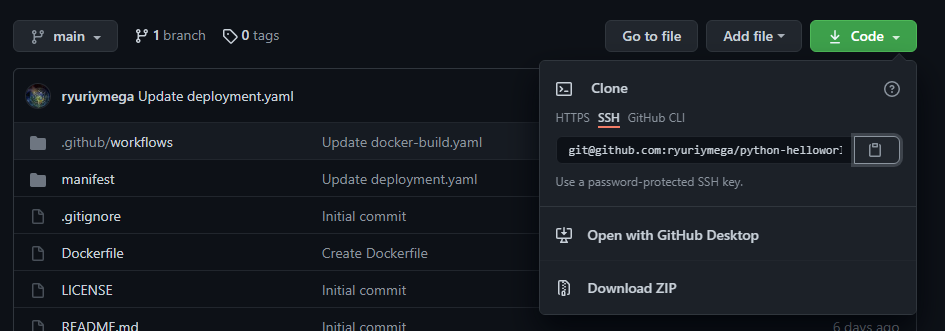
don't forget to add your ssh in the github security settings

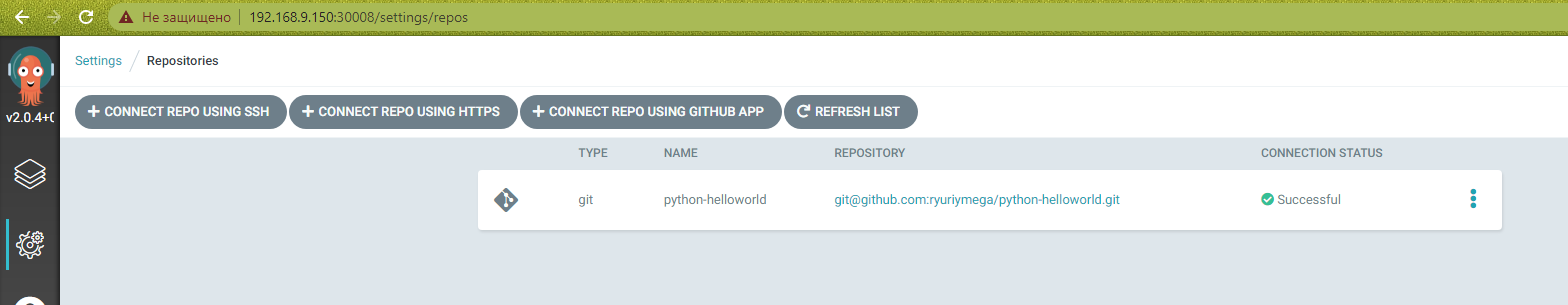


<https://docs.github.com/en/github/authenticating-to-github/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>



don't forget to copy ssh for repository settings and use in your repository





after all steps your argocd can connect with your secure github account  
all ready

