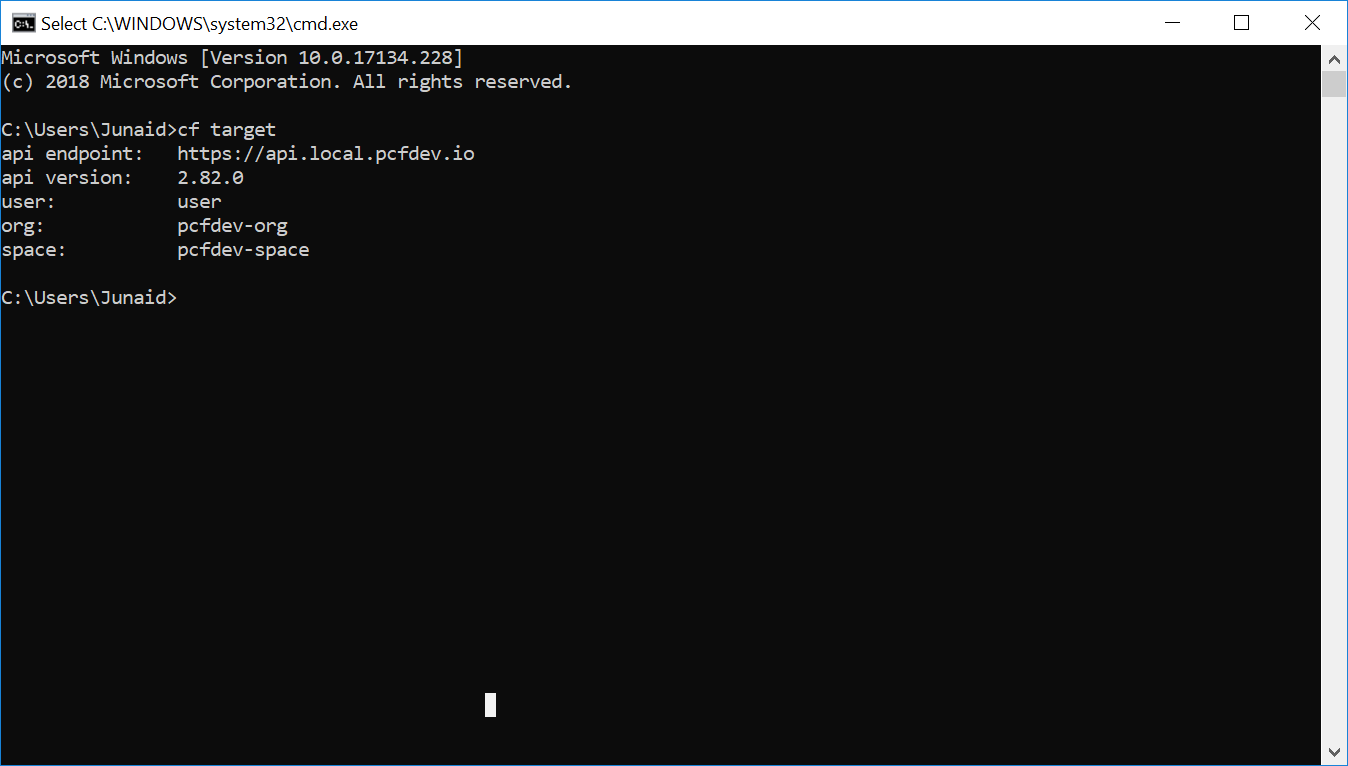
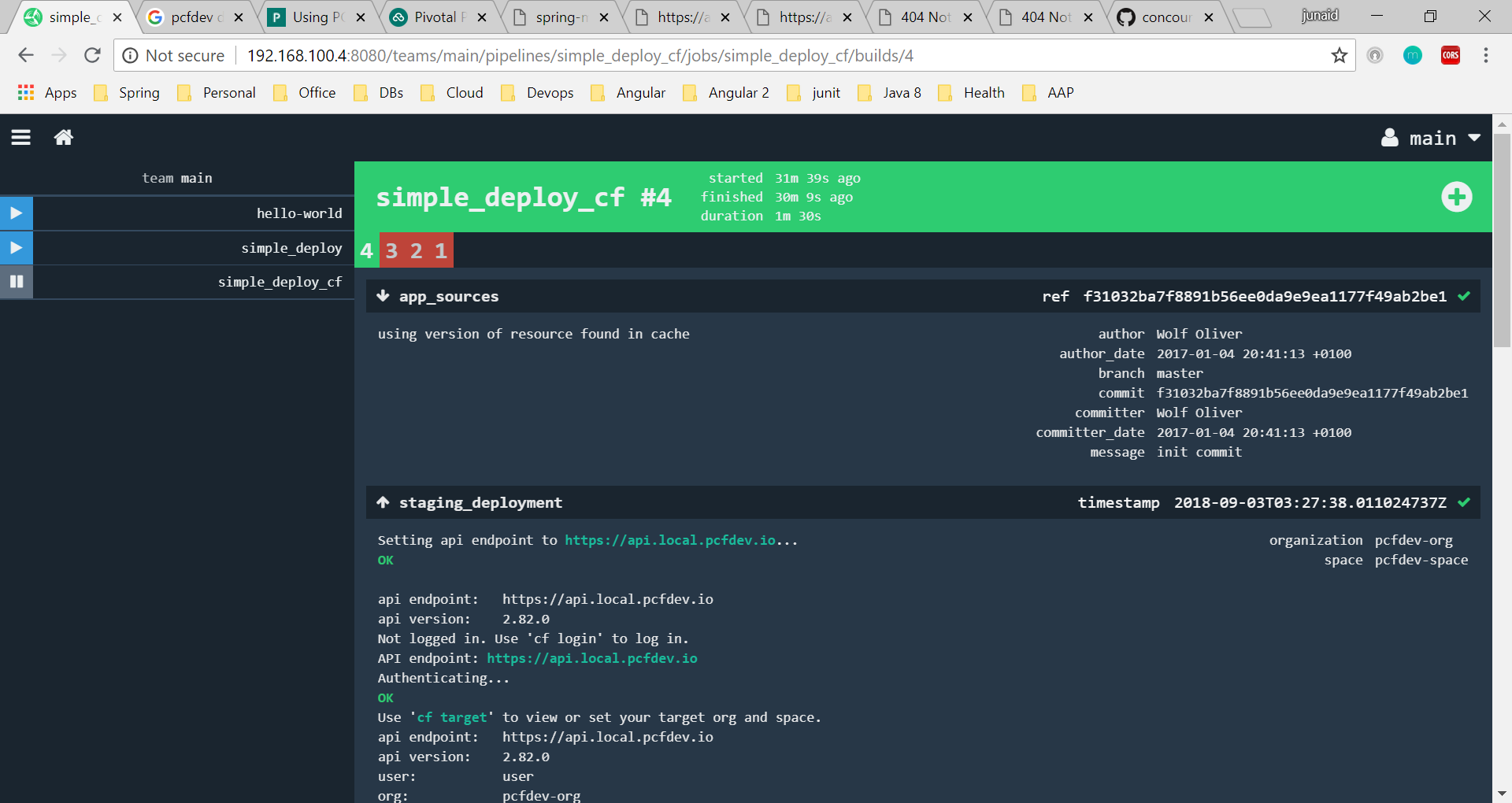
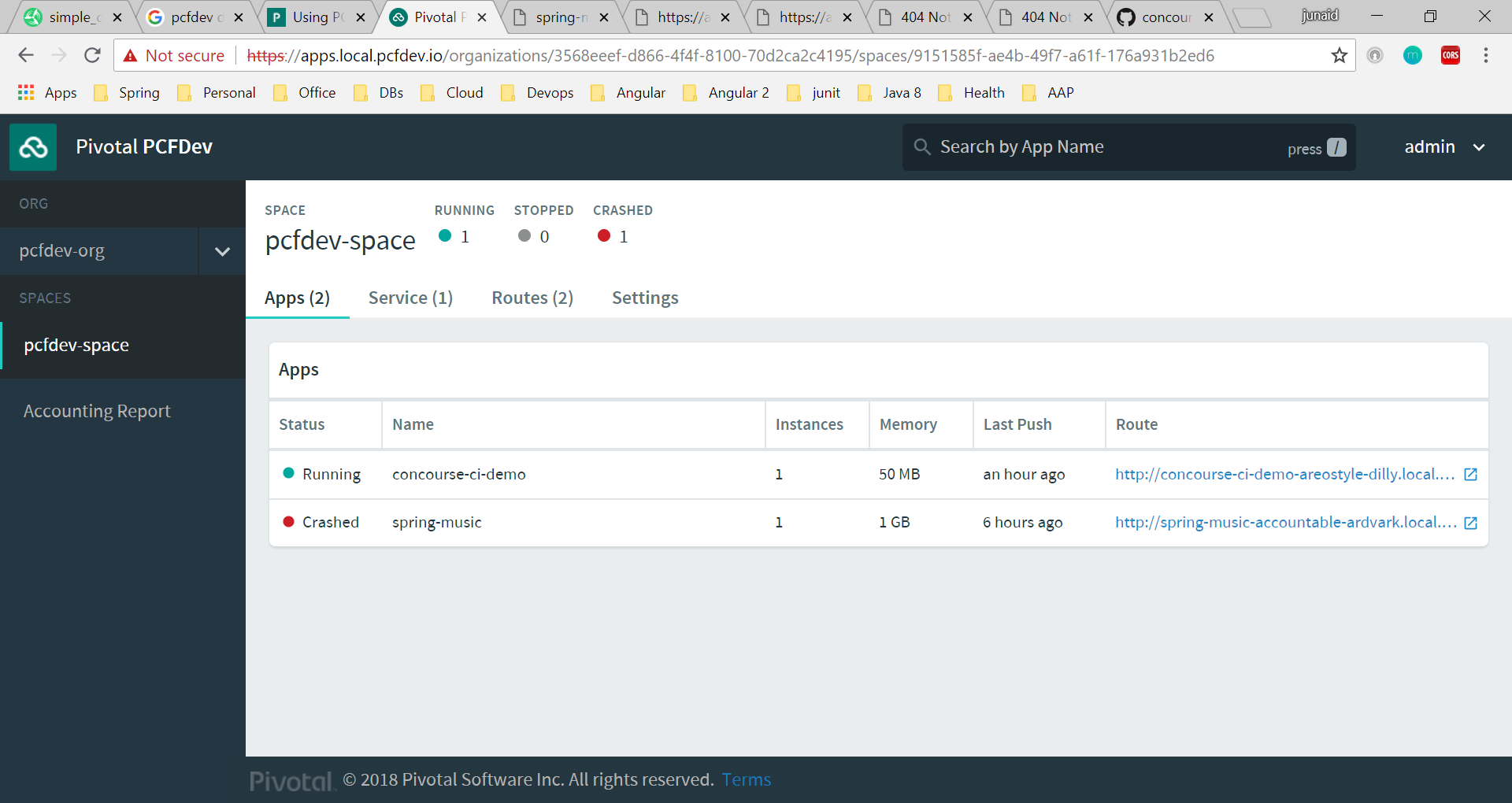


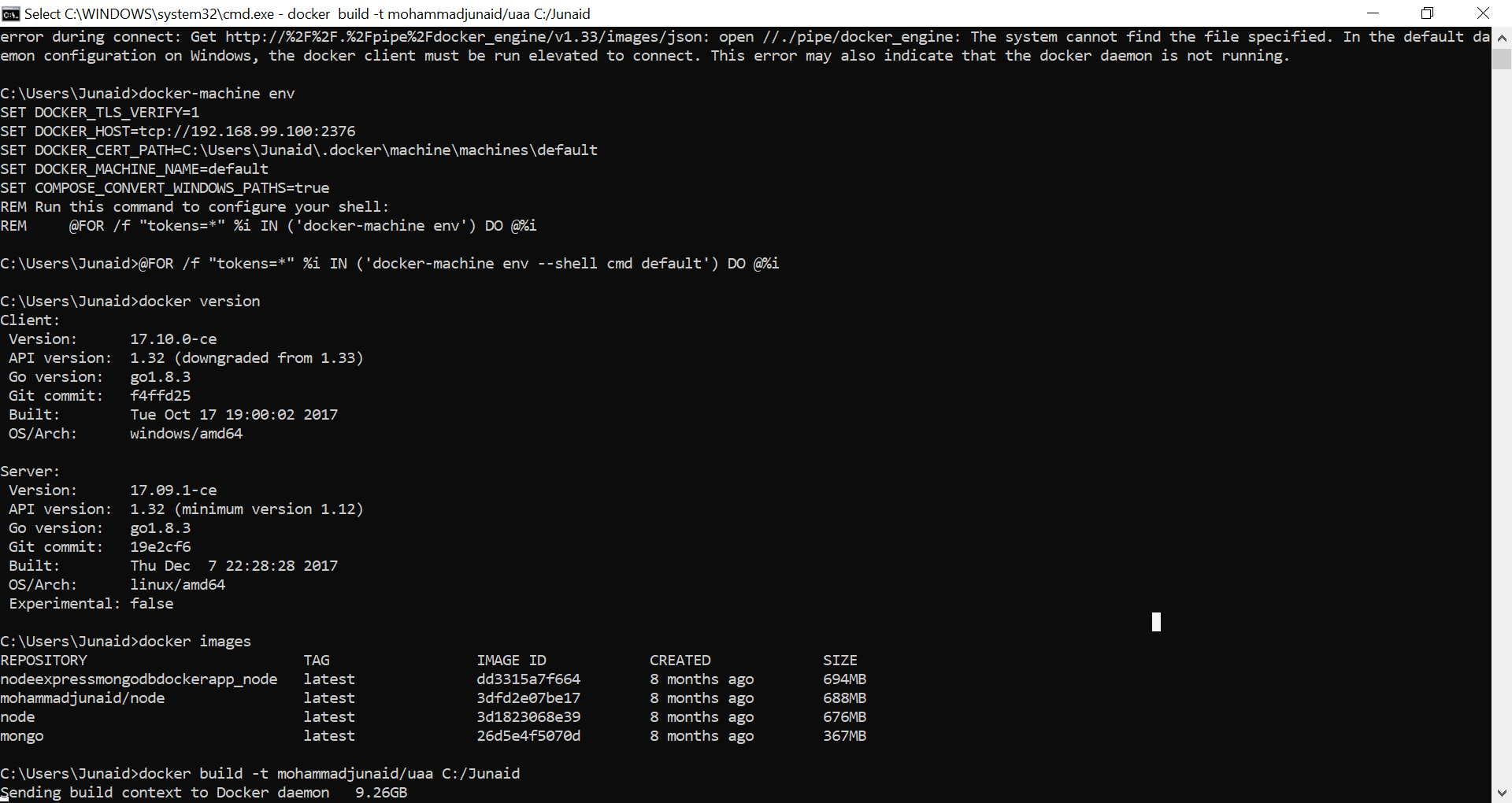
Use cf target



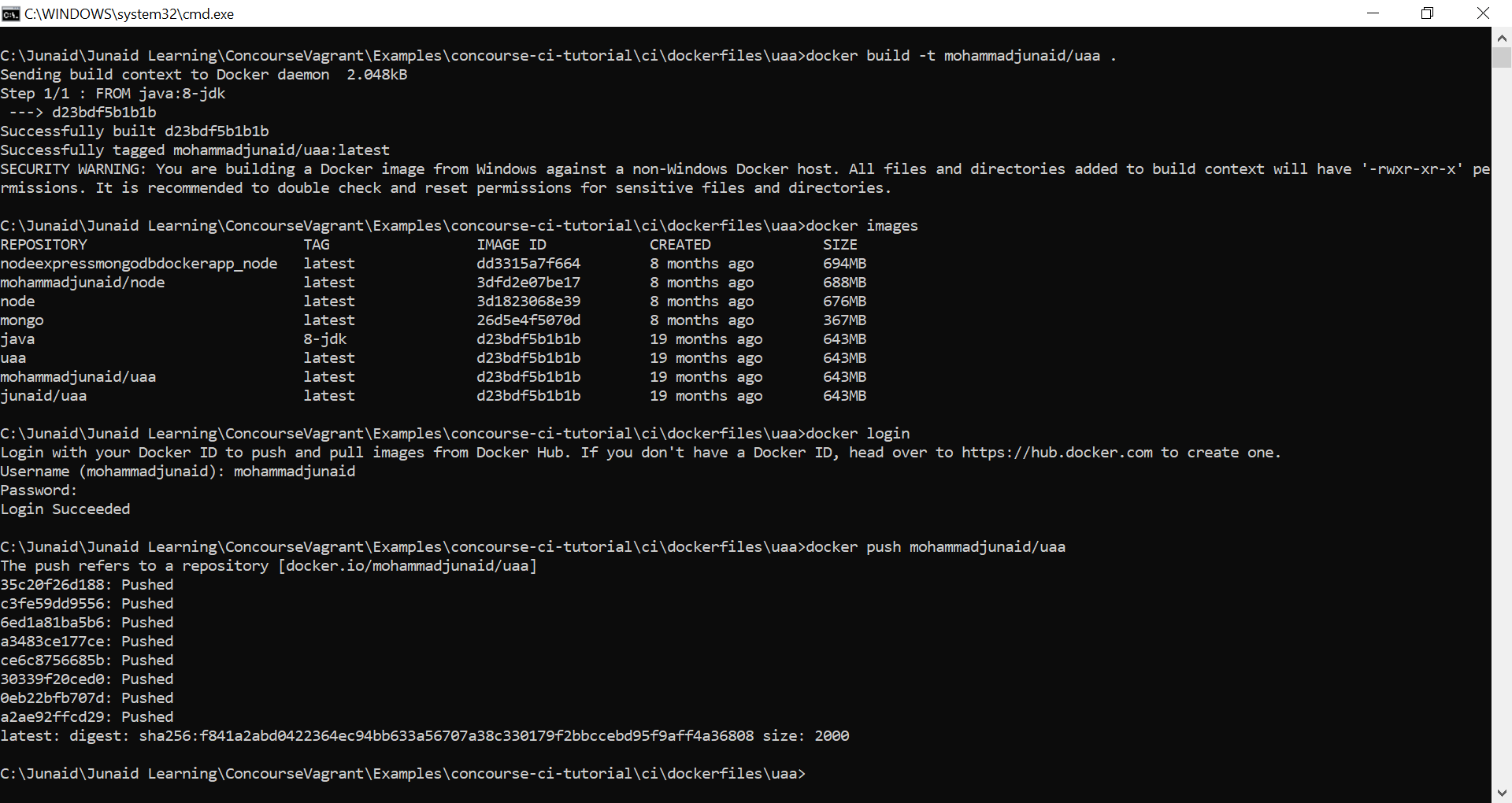




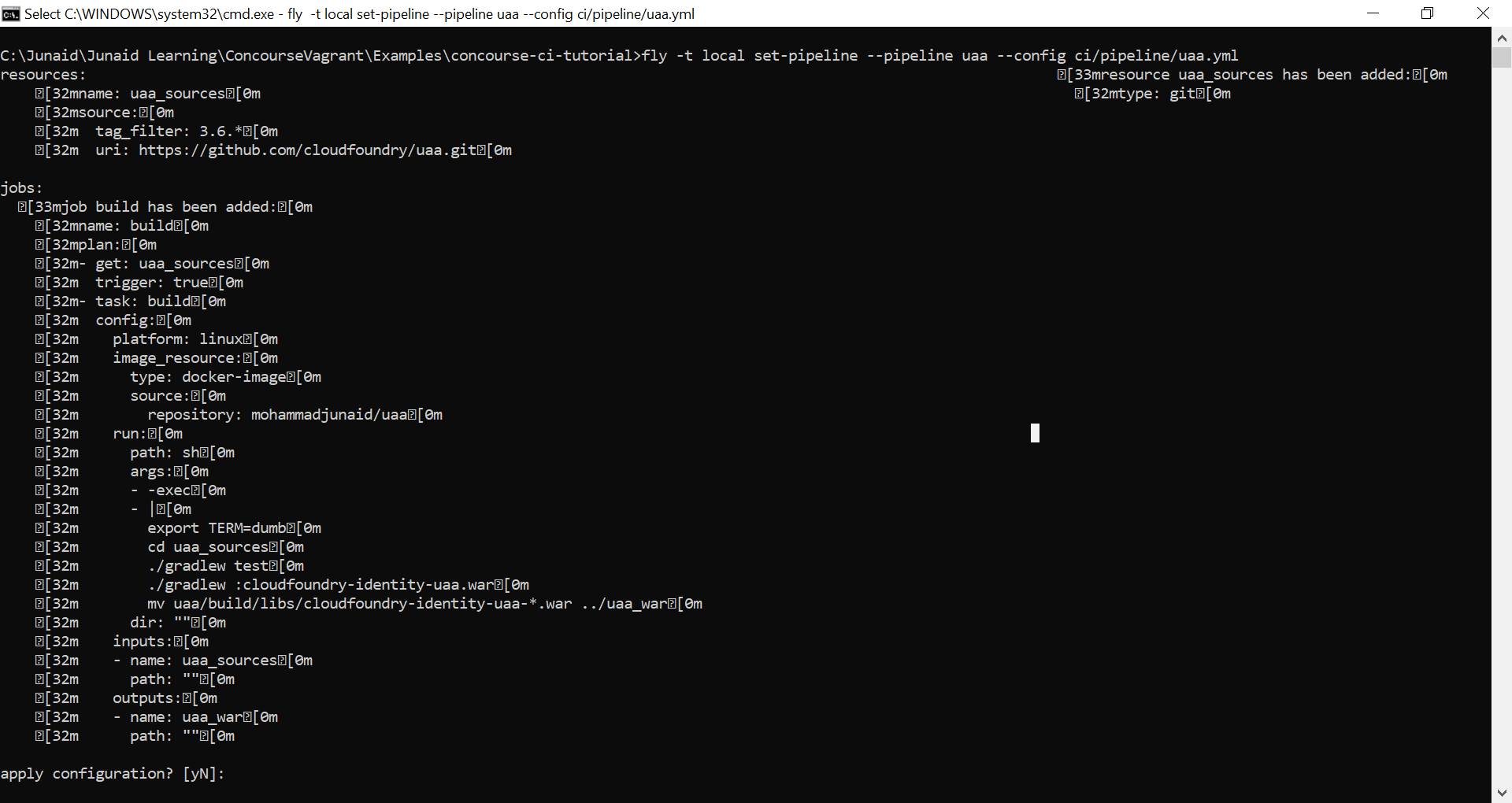
Pipe the first service to production (Java App)

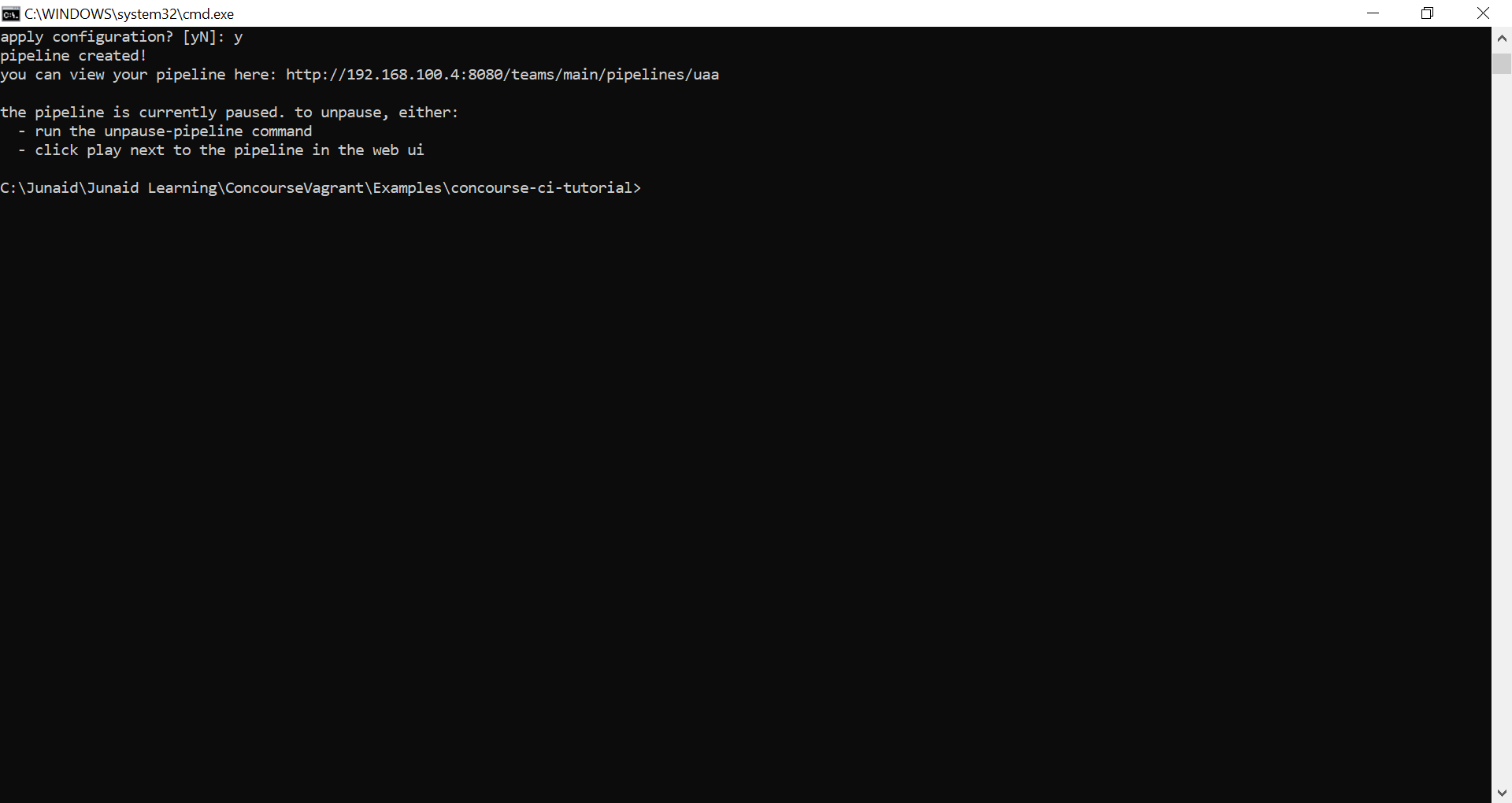


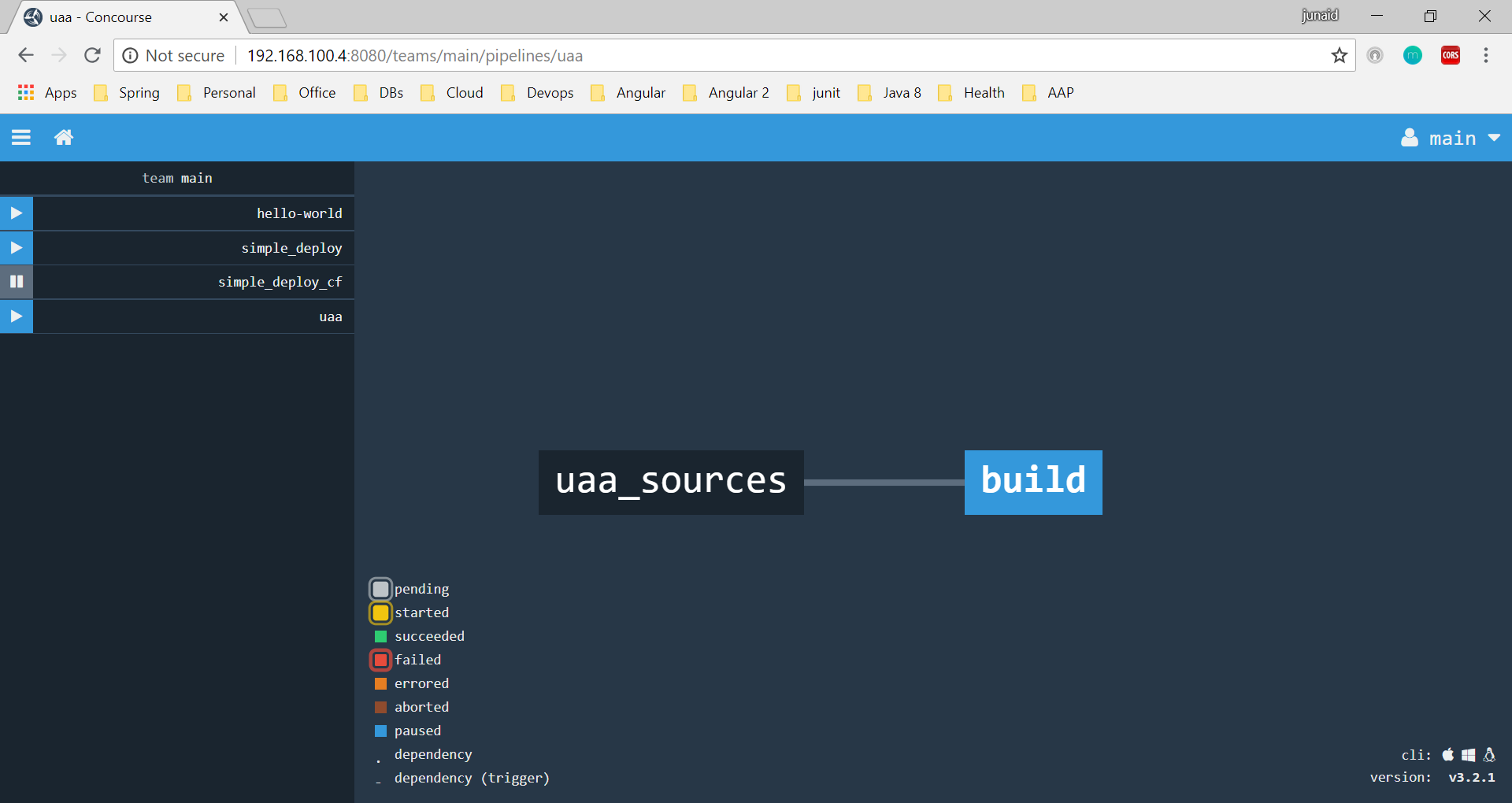
https://blog.codeship.com/using-docker-push-to-publish-images-to-dockerhub/



Step 3: Create concourse job to build the UAA and run the unit tests

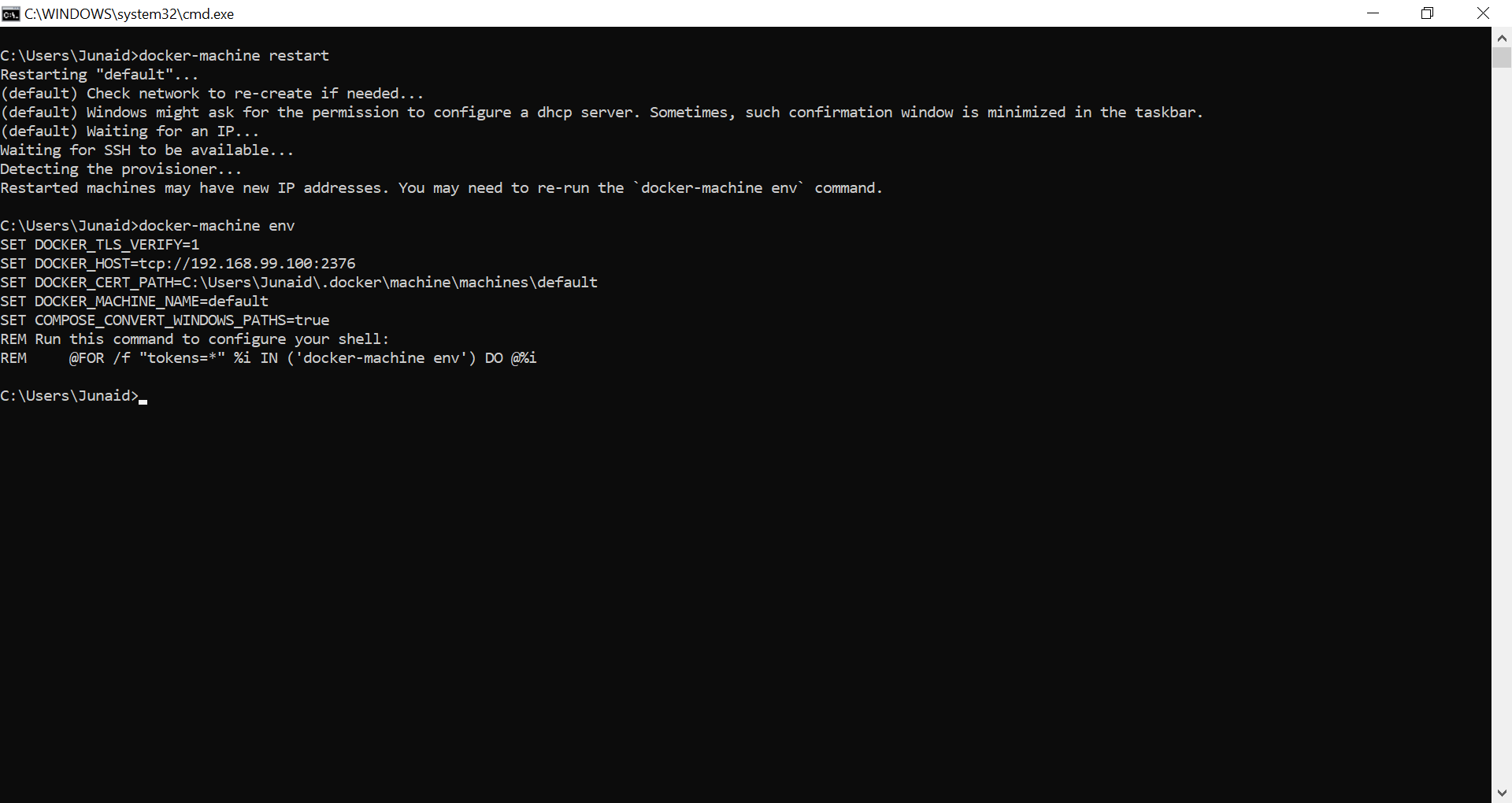


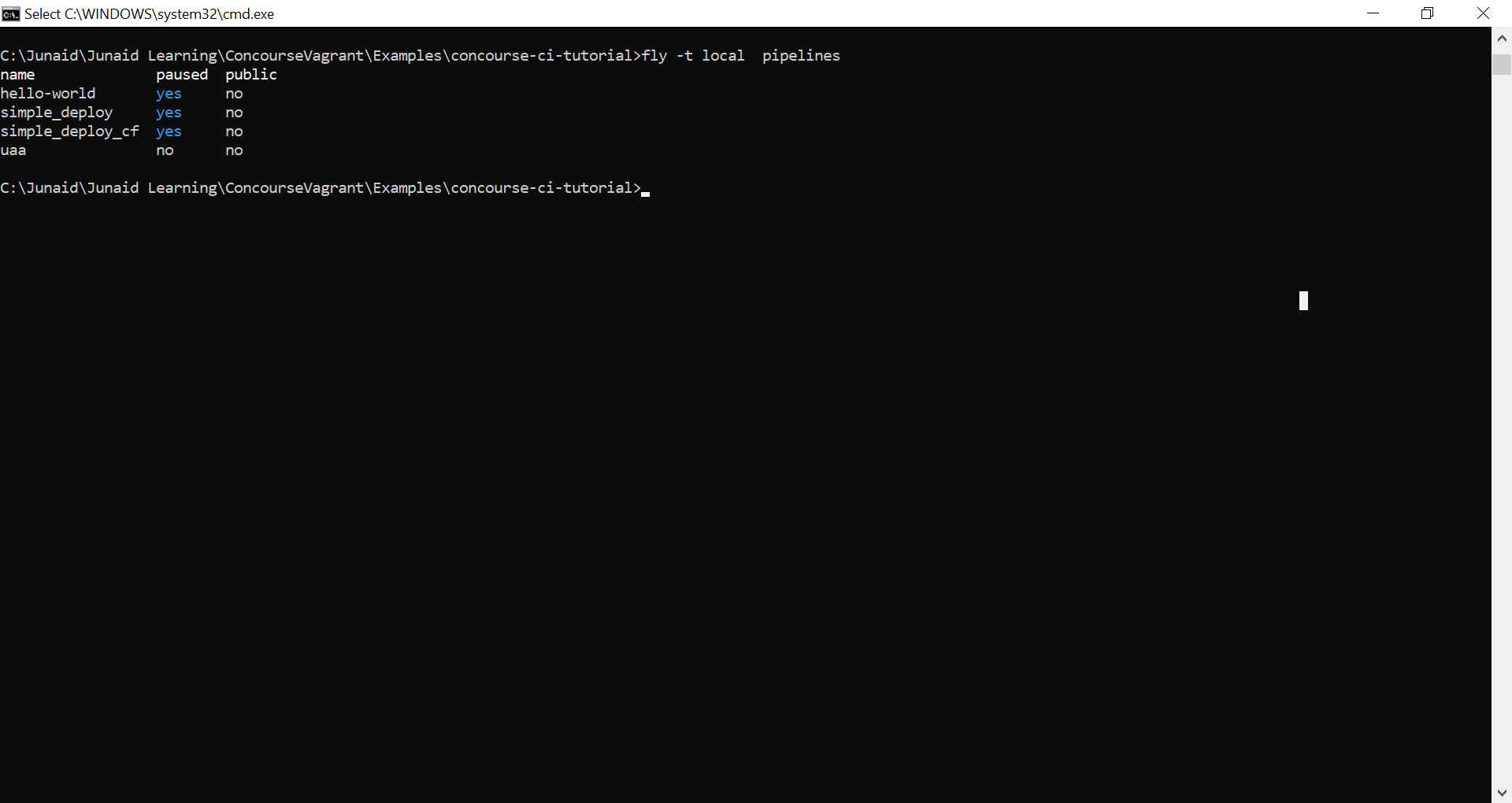




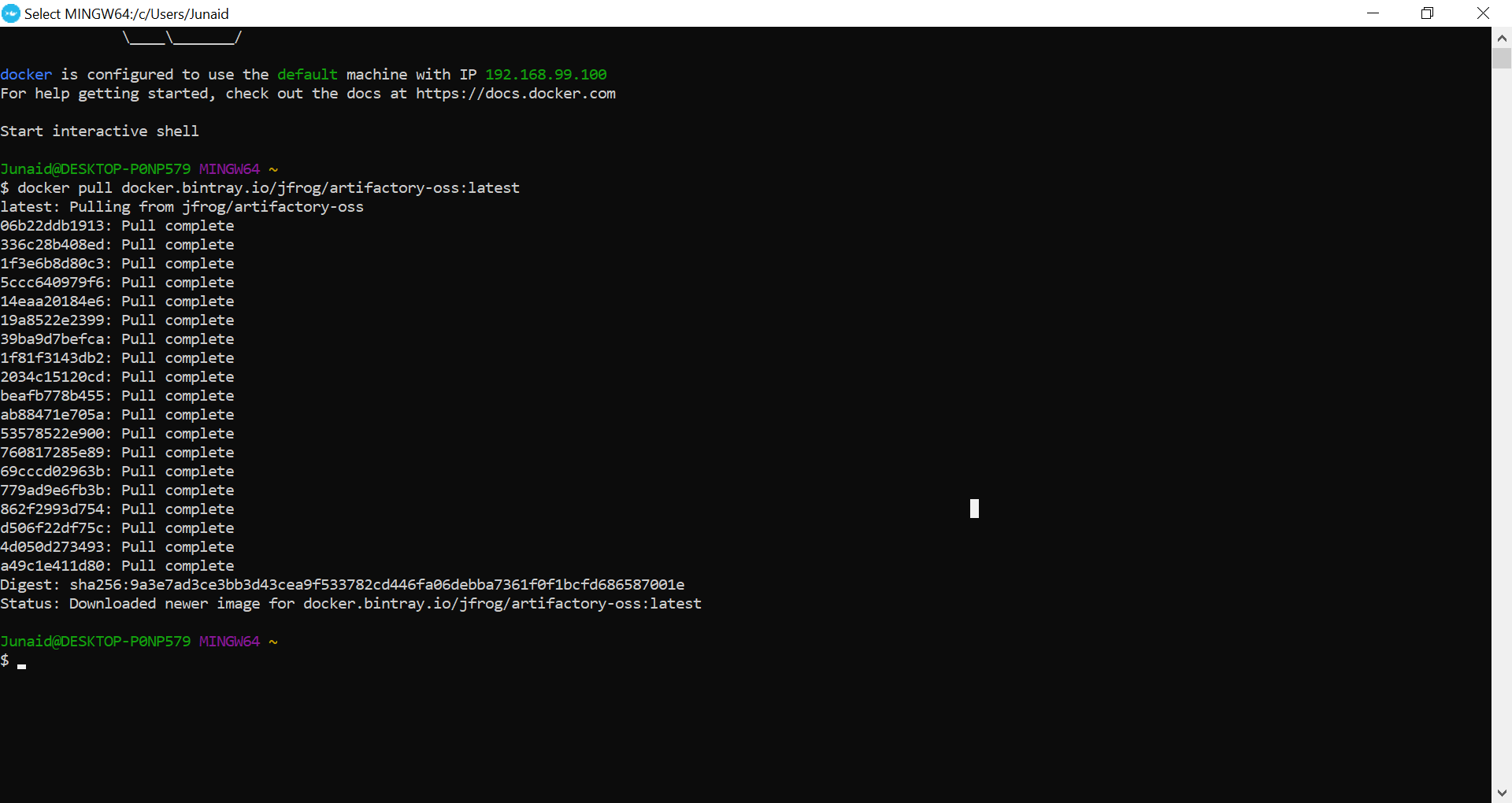
Step 4a: Setup local Artifactory server

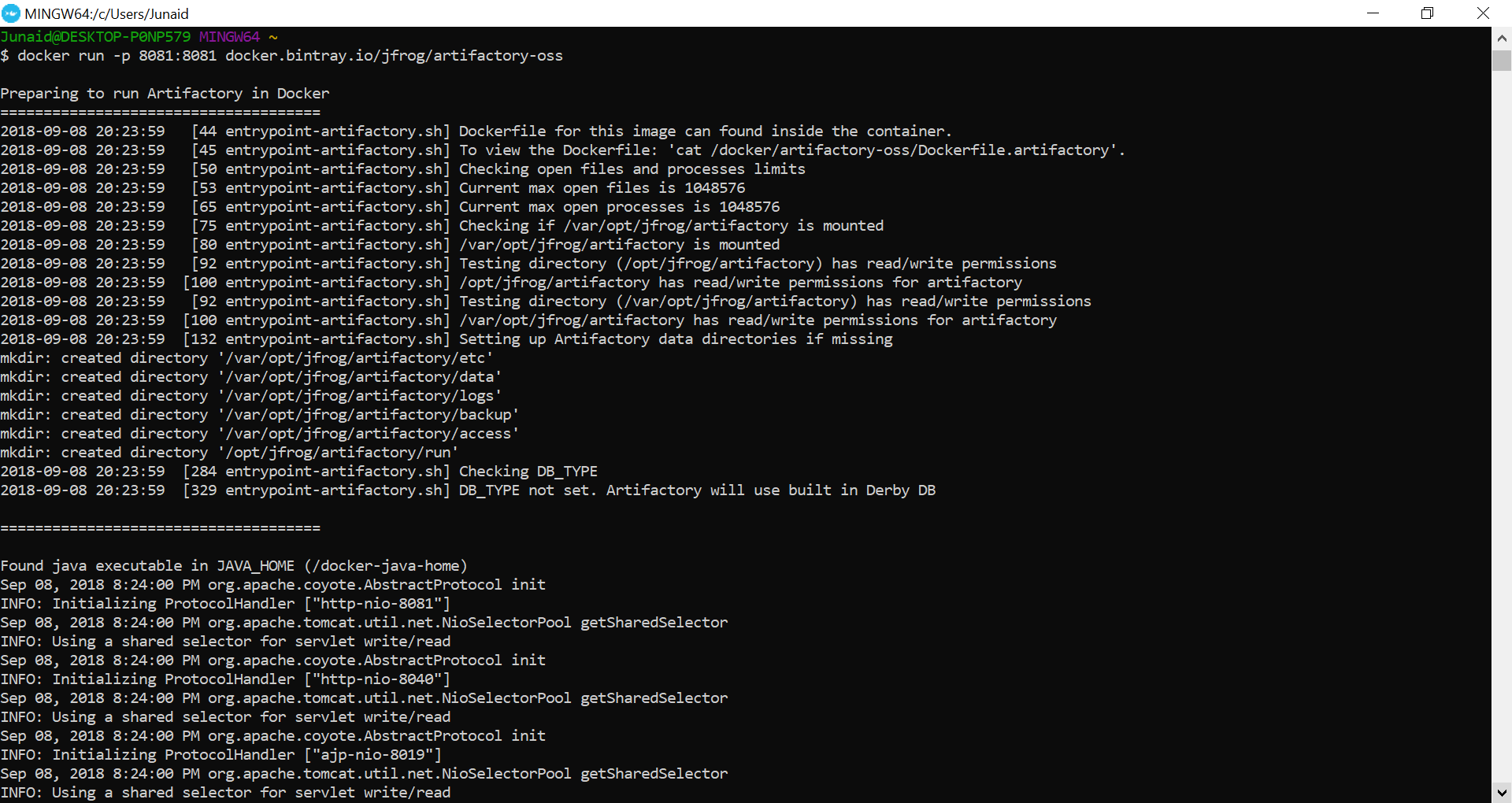
Start Docker engine

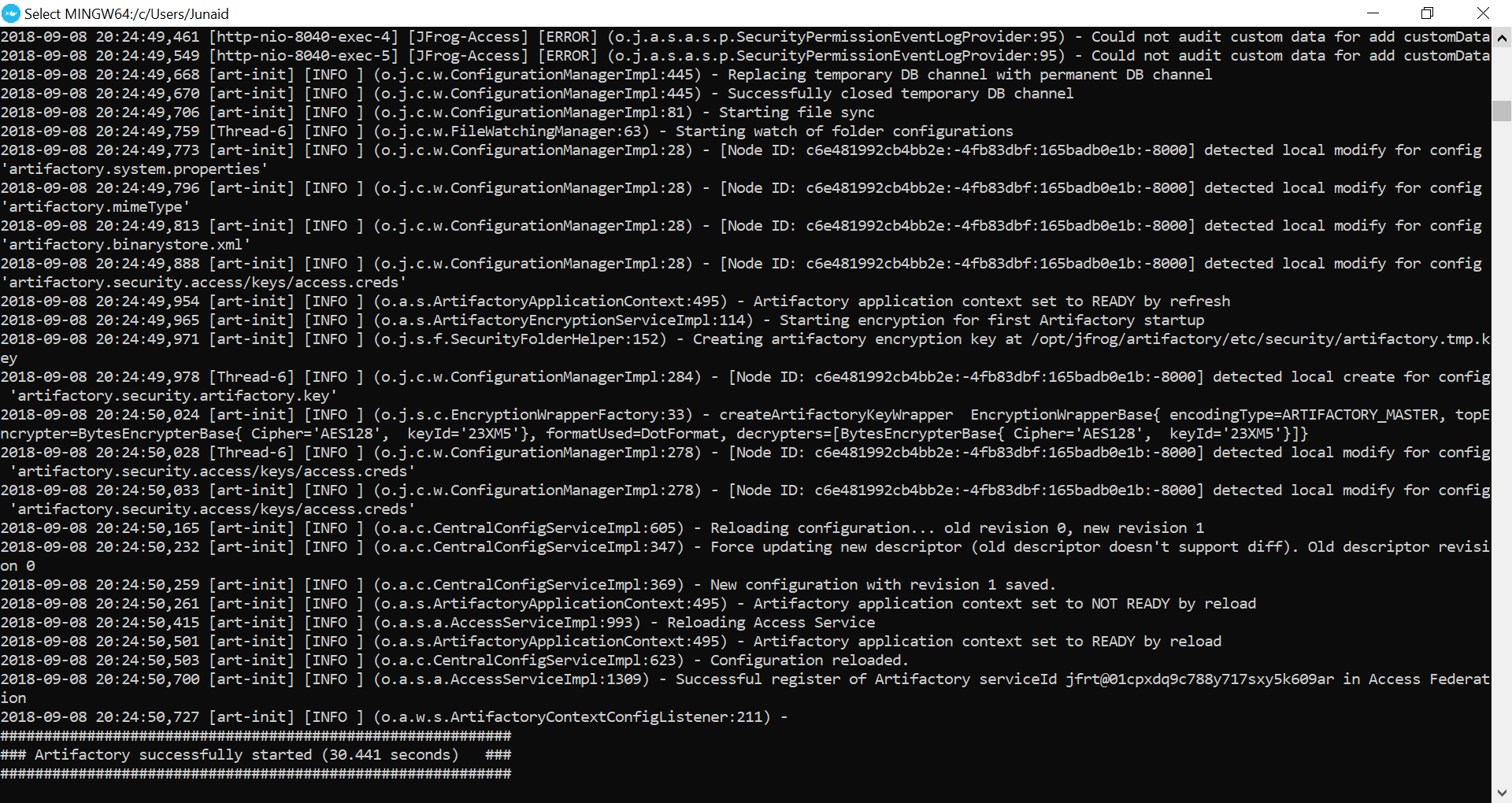




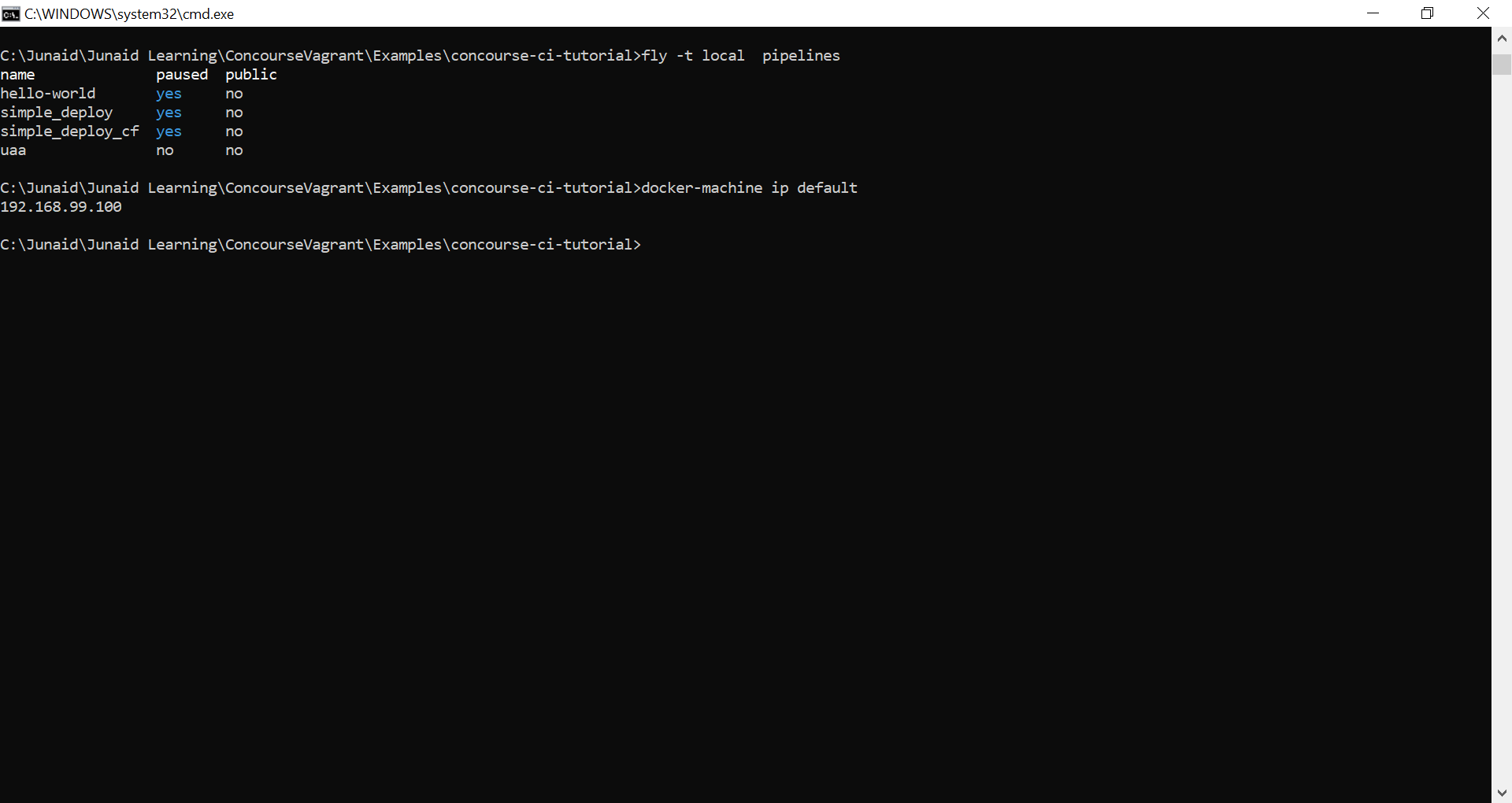
There is a Docker image for Artifactory to get it running quickly. Start your Docker engine and execute:



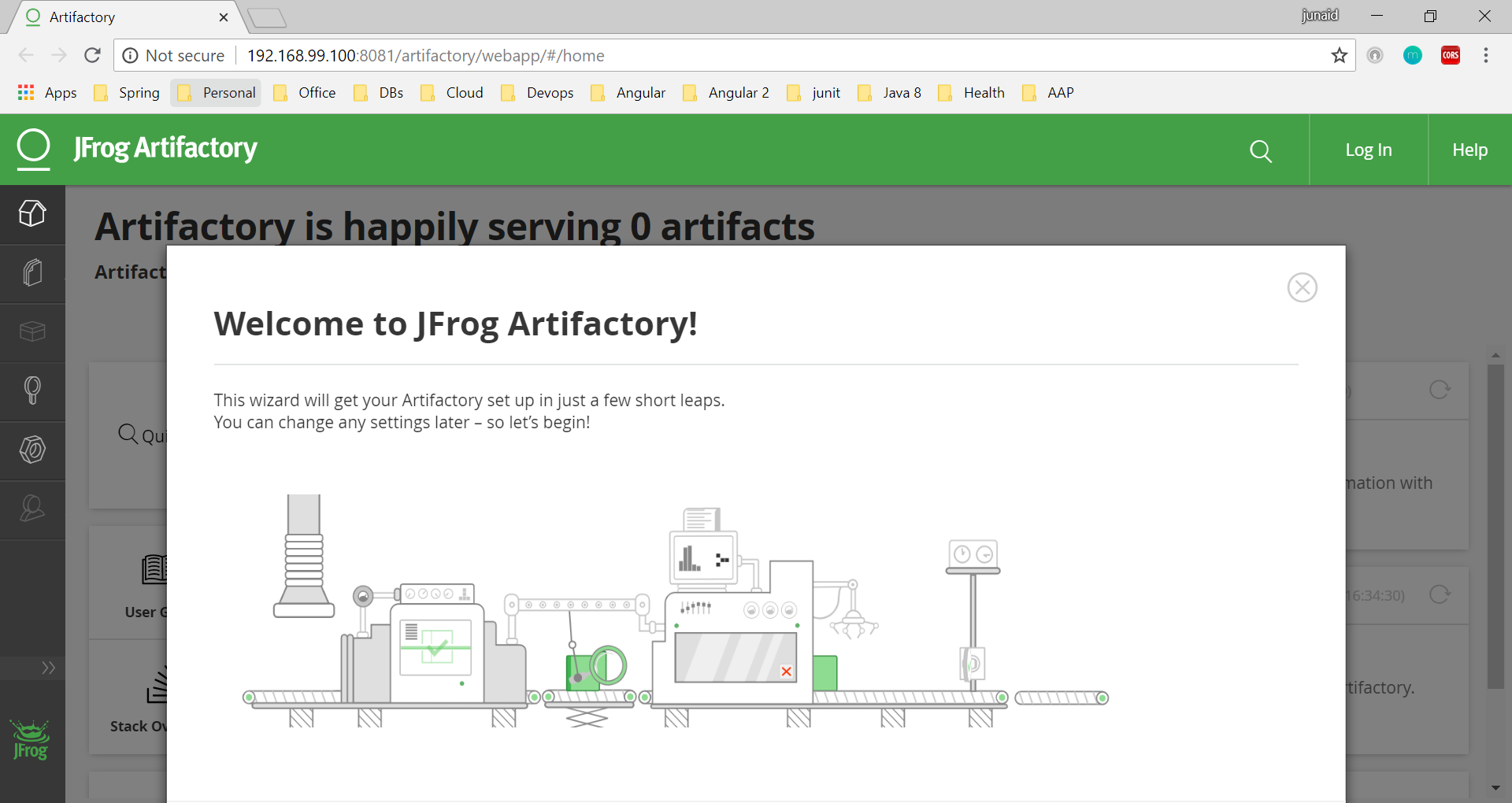


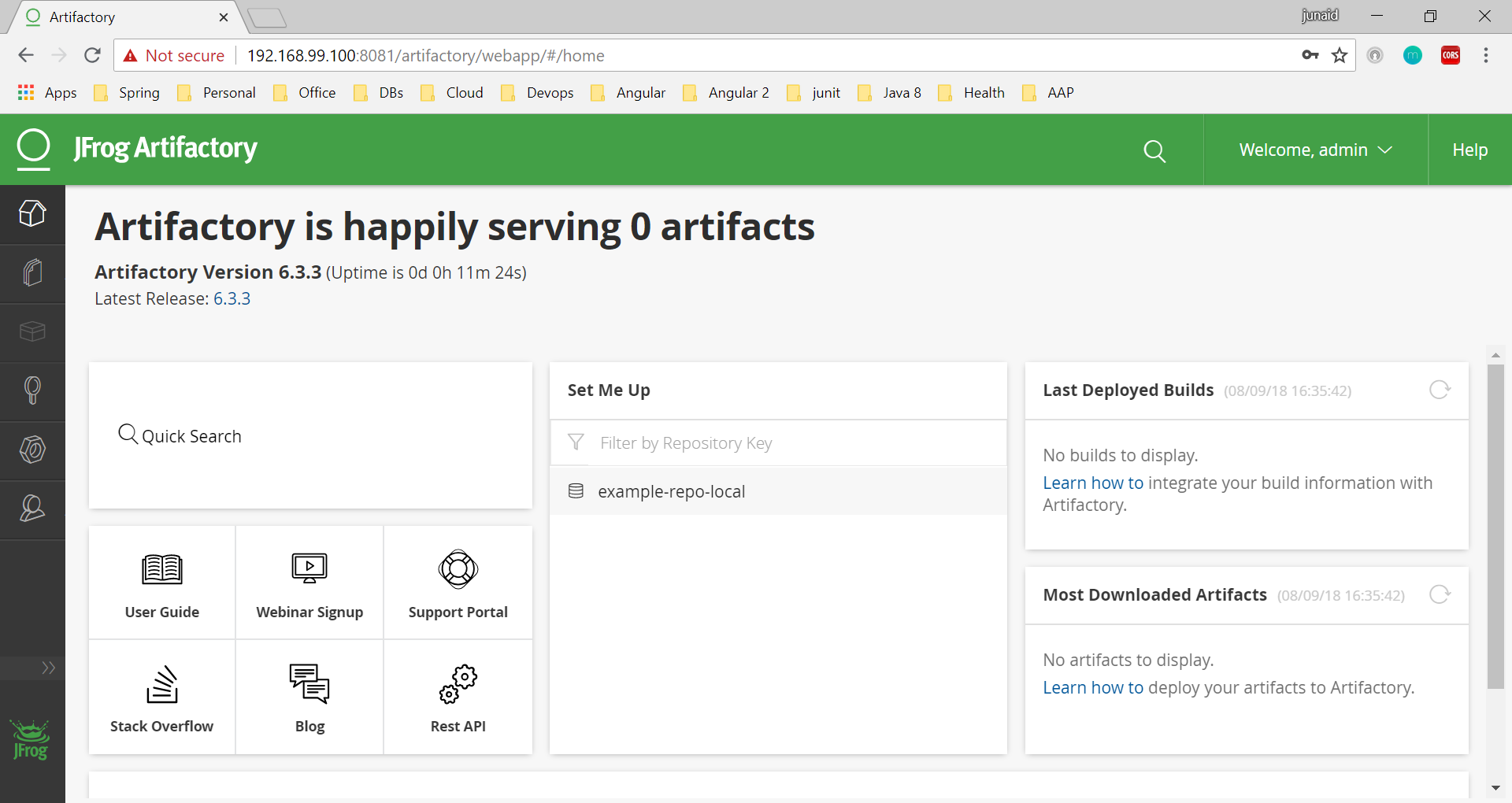


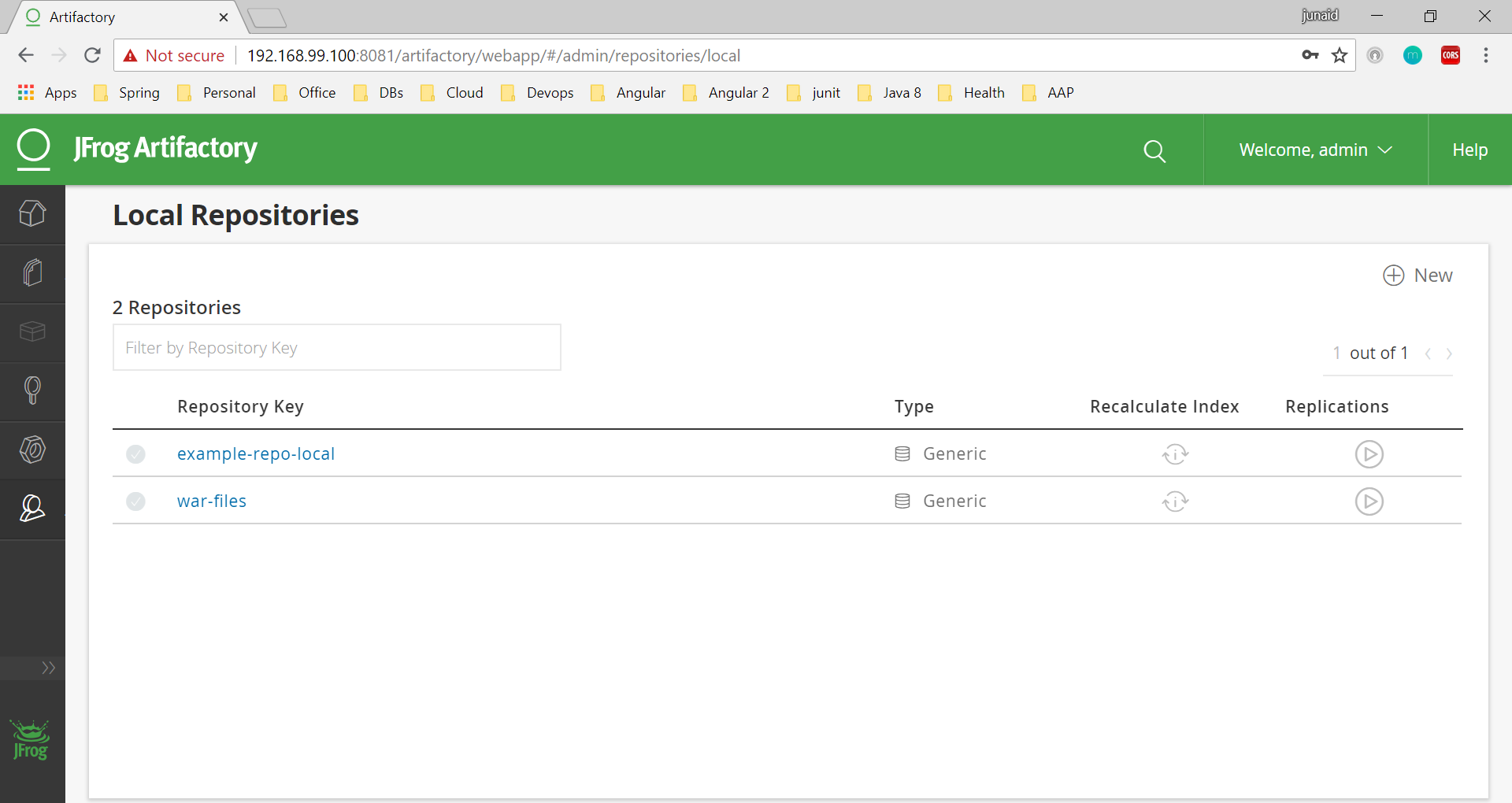
Finding out docker engine ip

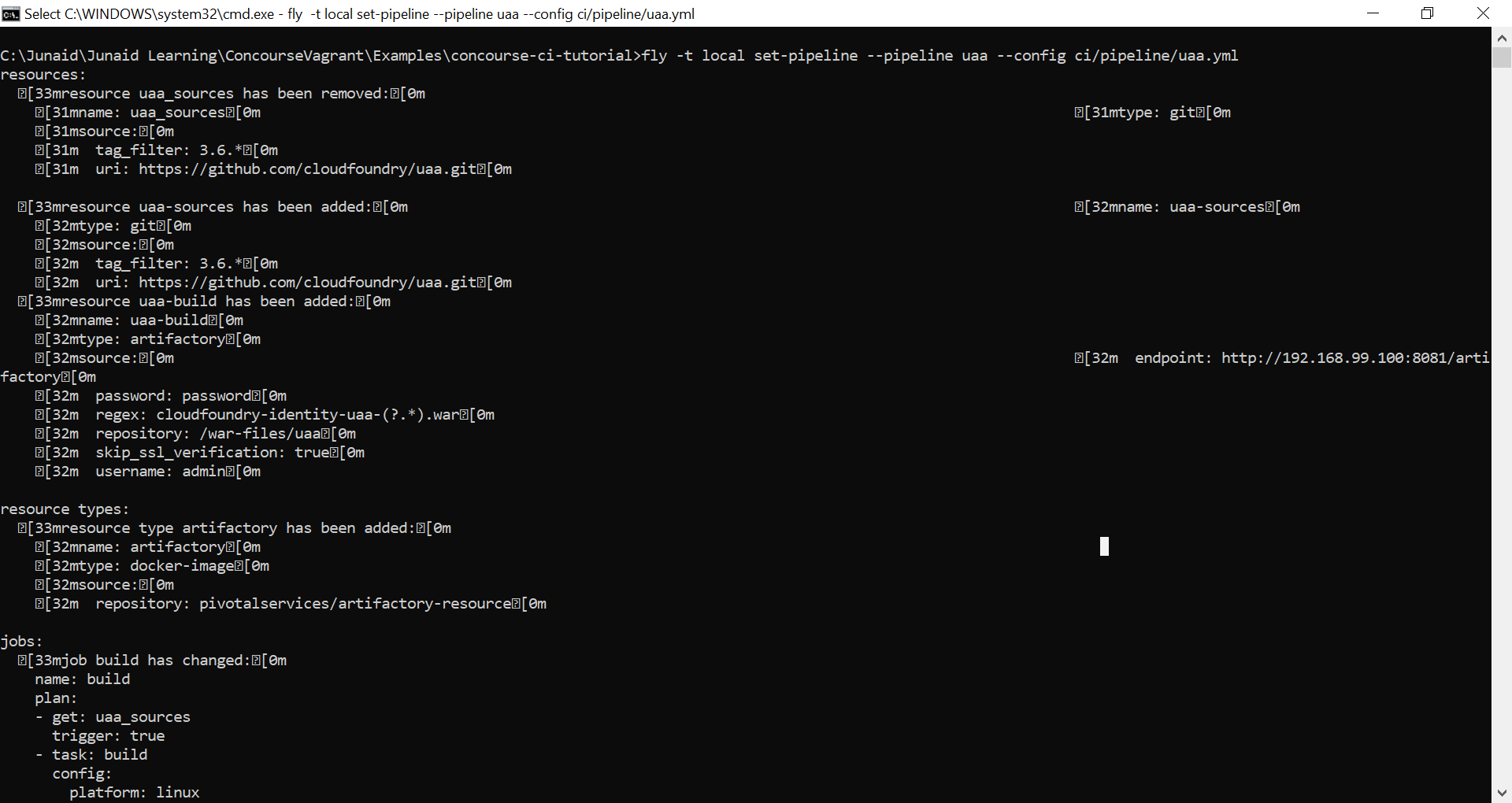


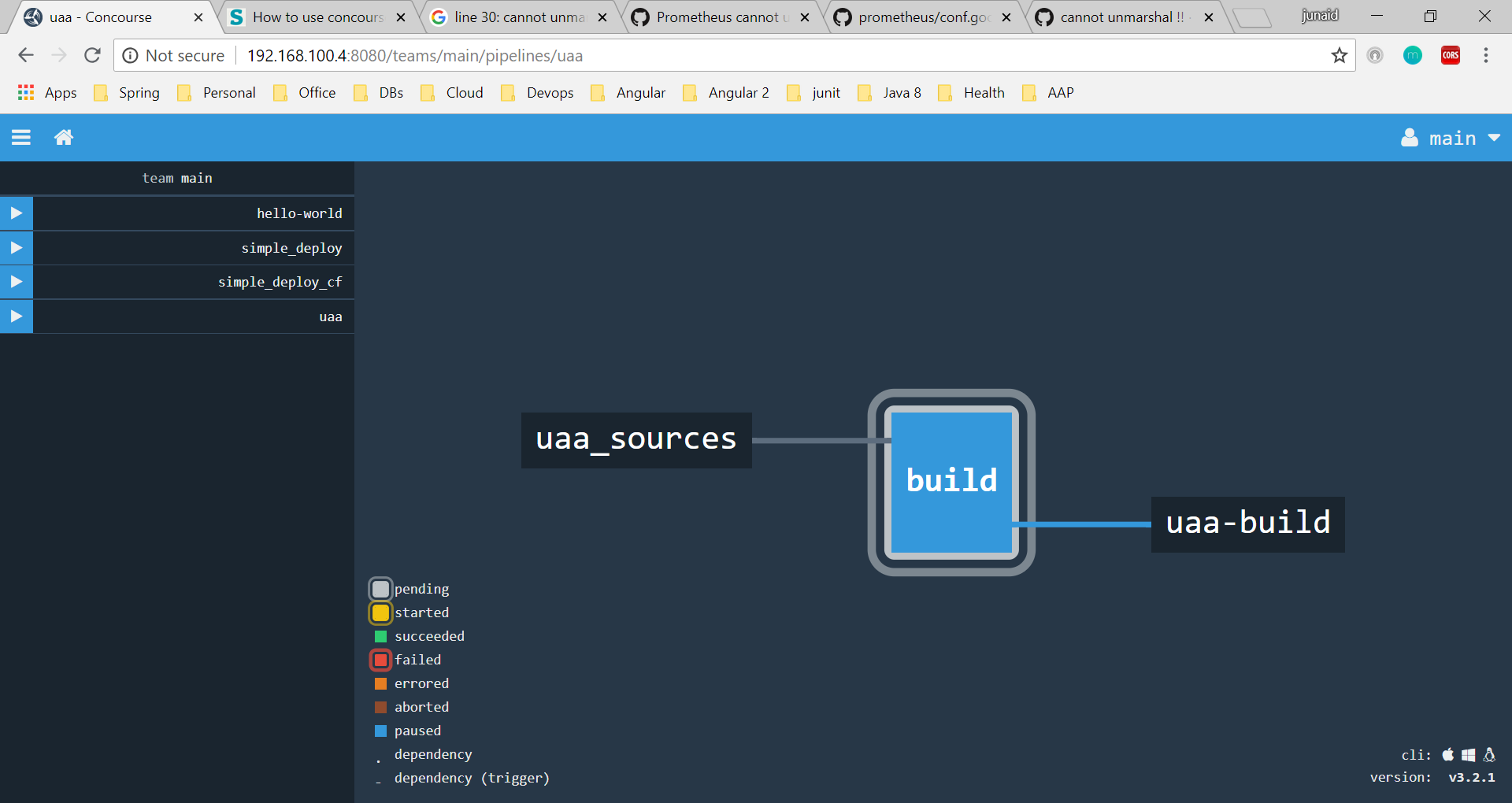
If the docker machine IP is 192.168.99.100 you can check whether Artifactory is running by opening http://192.168.99.100:8081 in your browser.



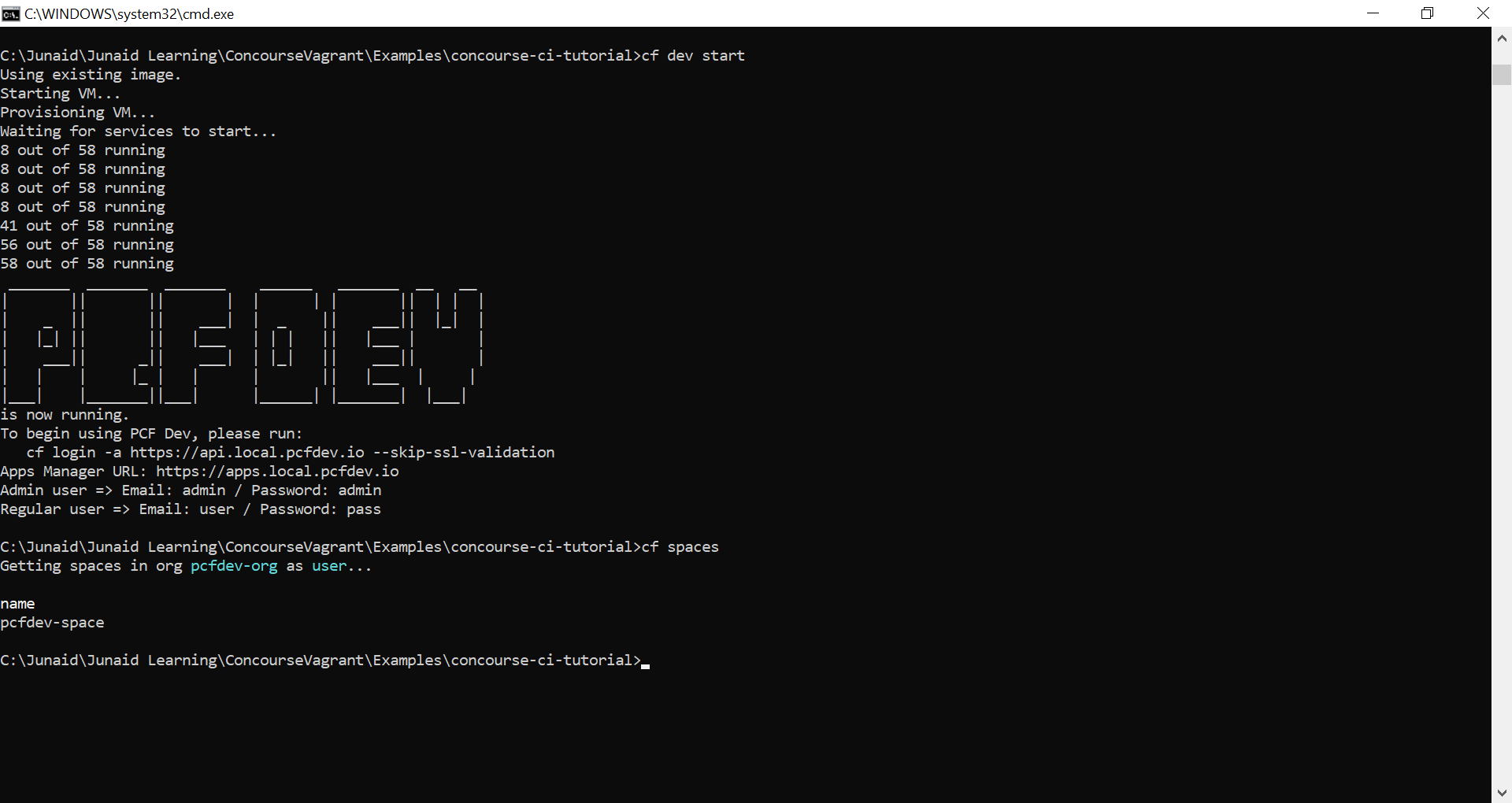


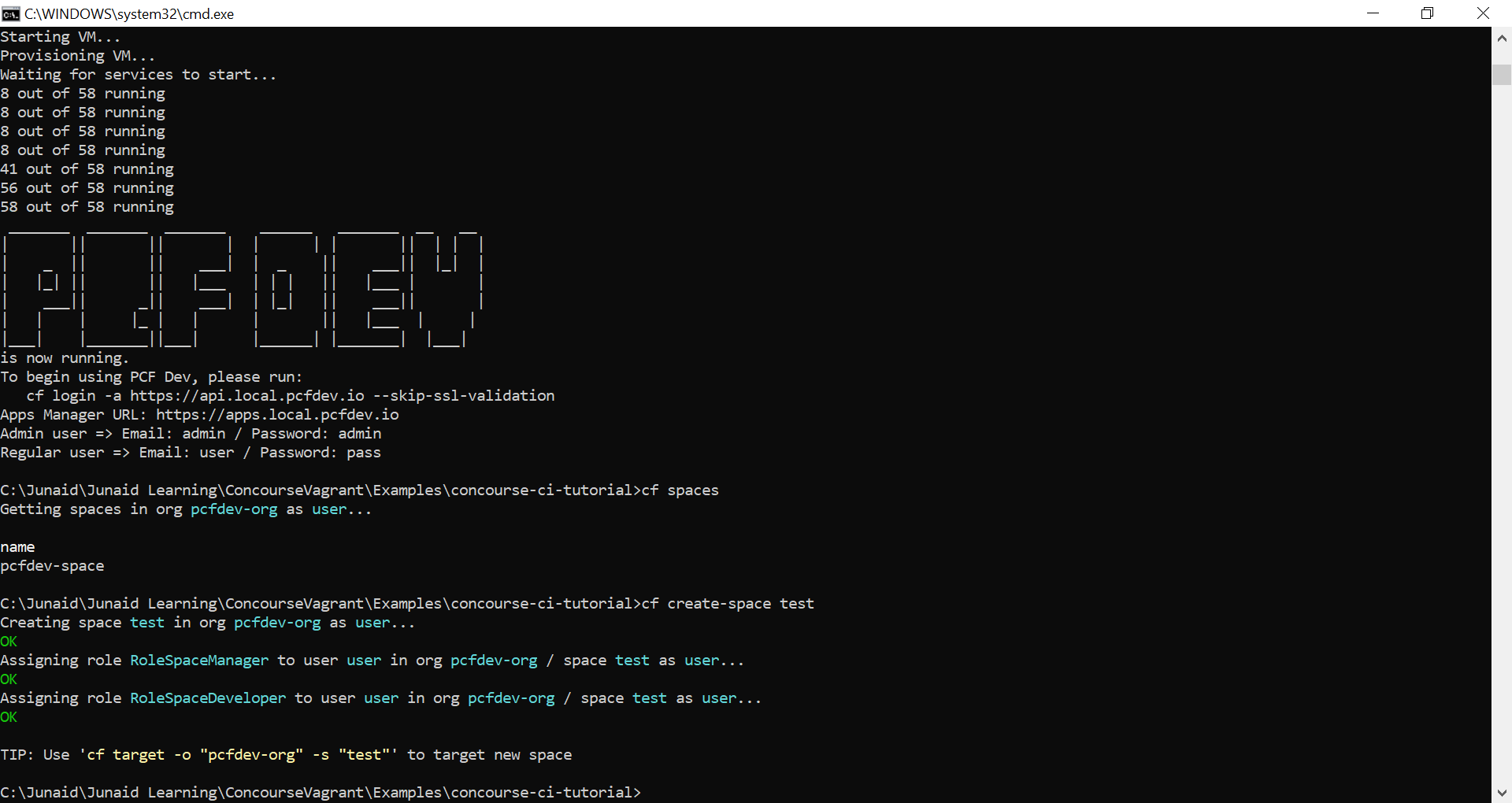


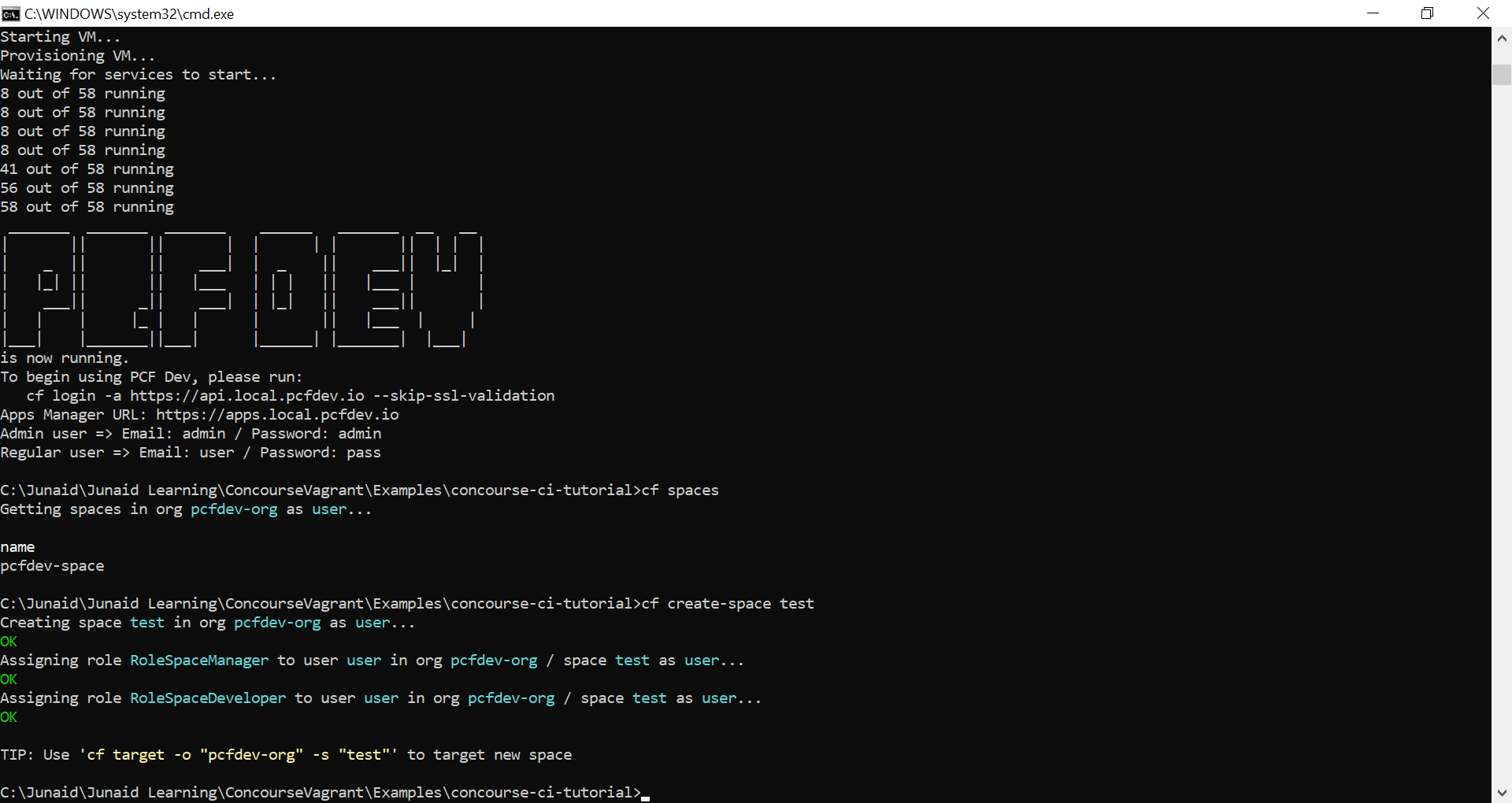


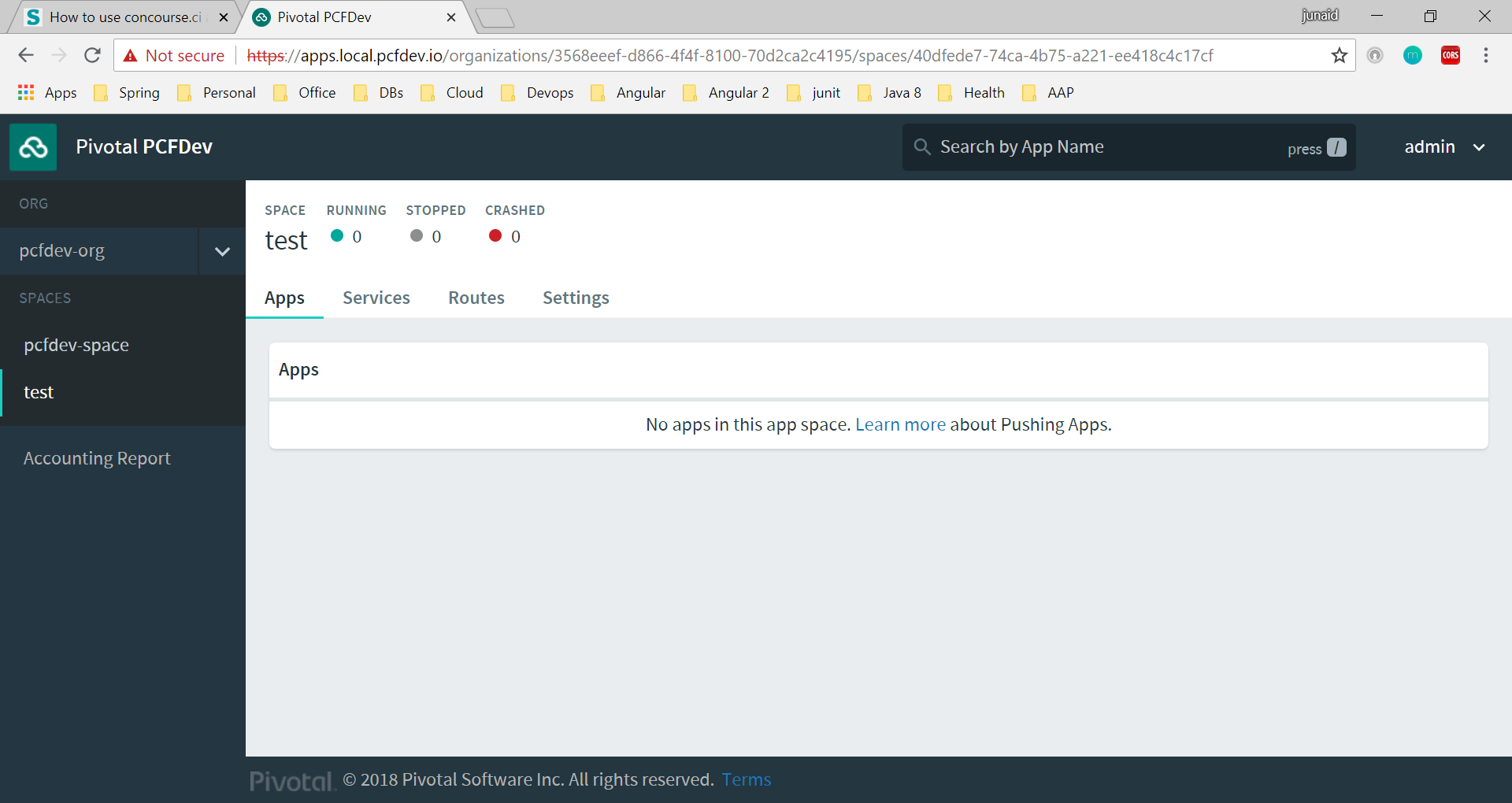


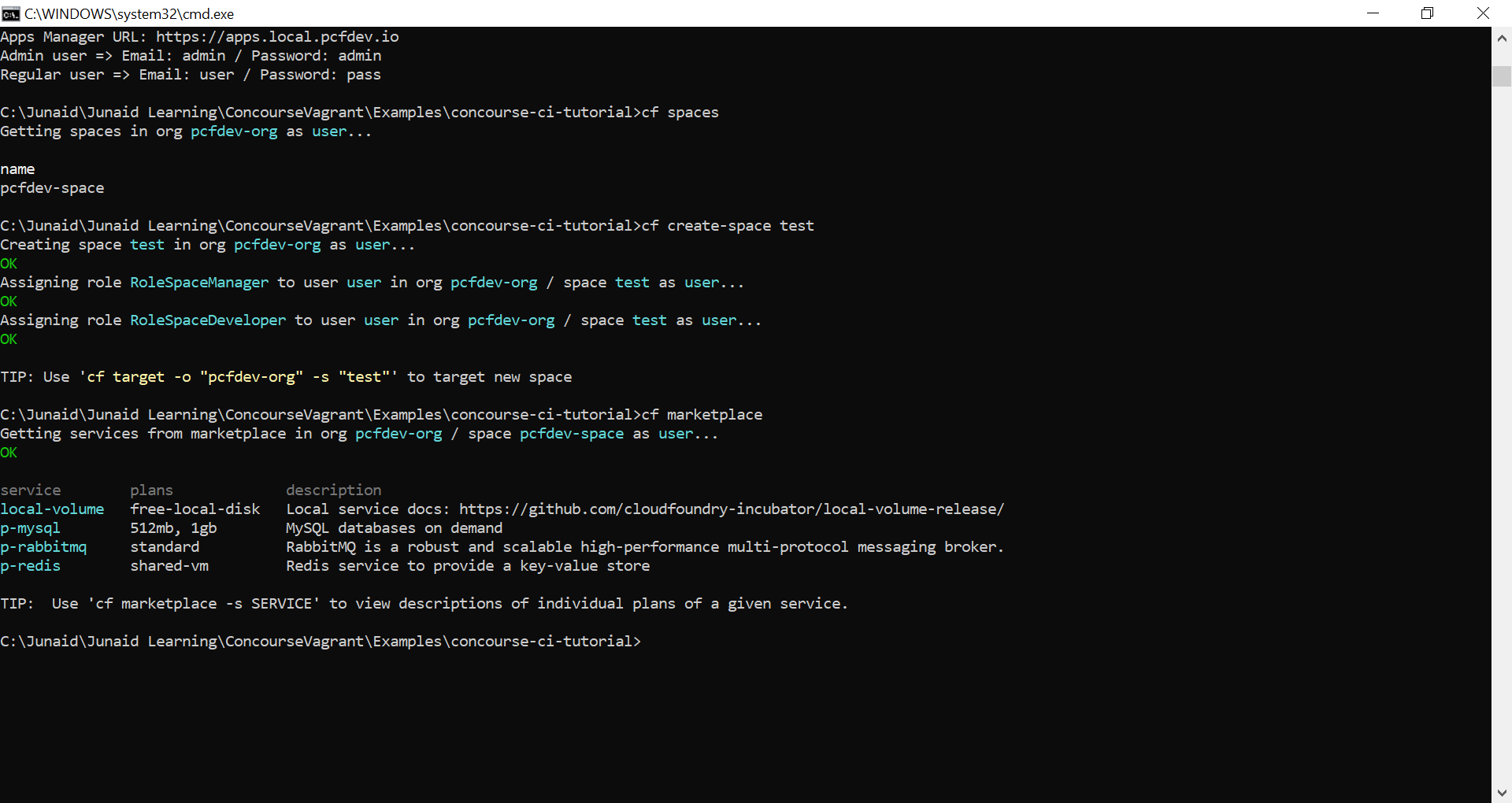
Step 5: Create RDBMS instances



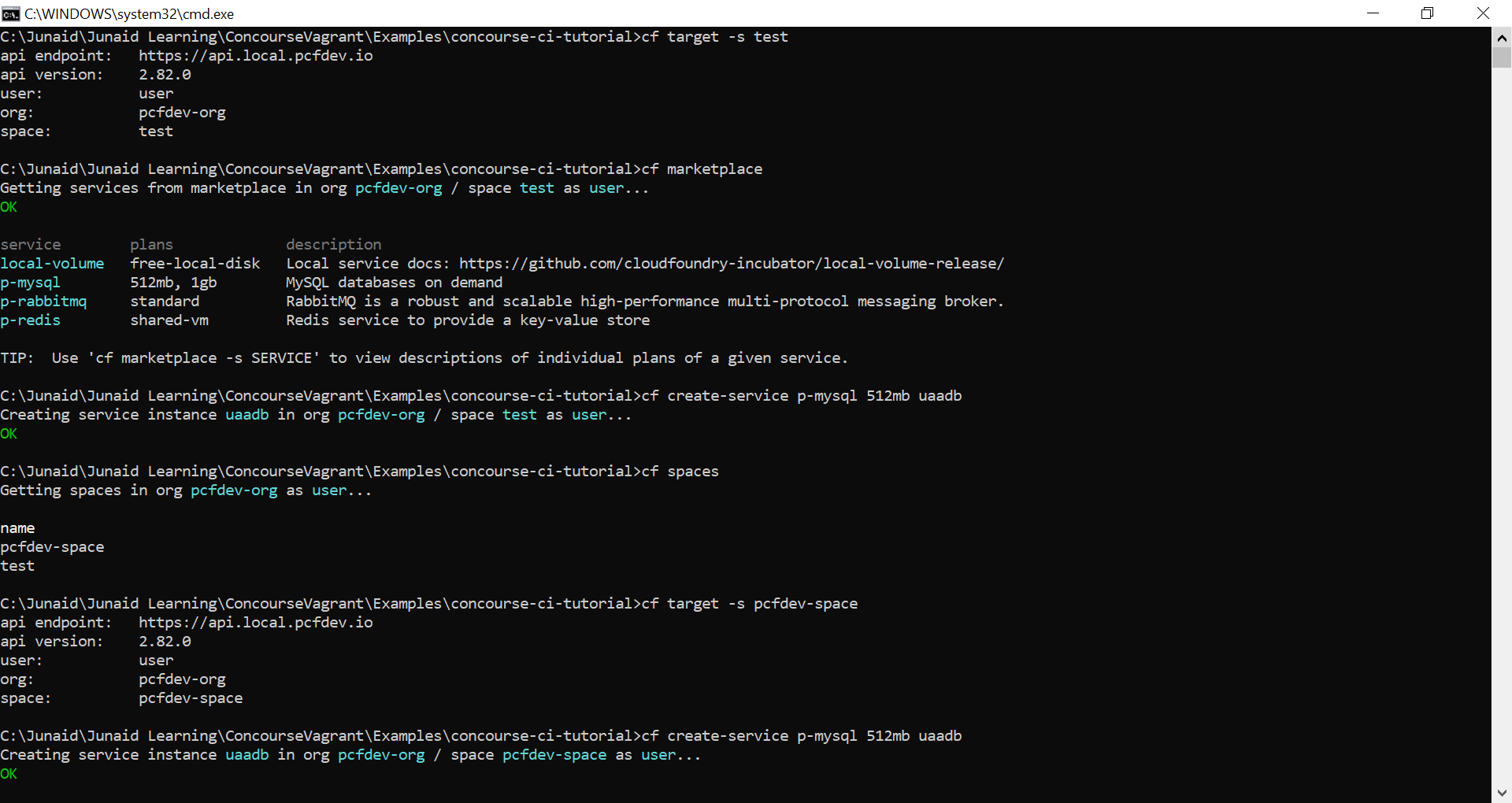


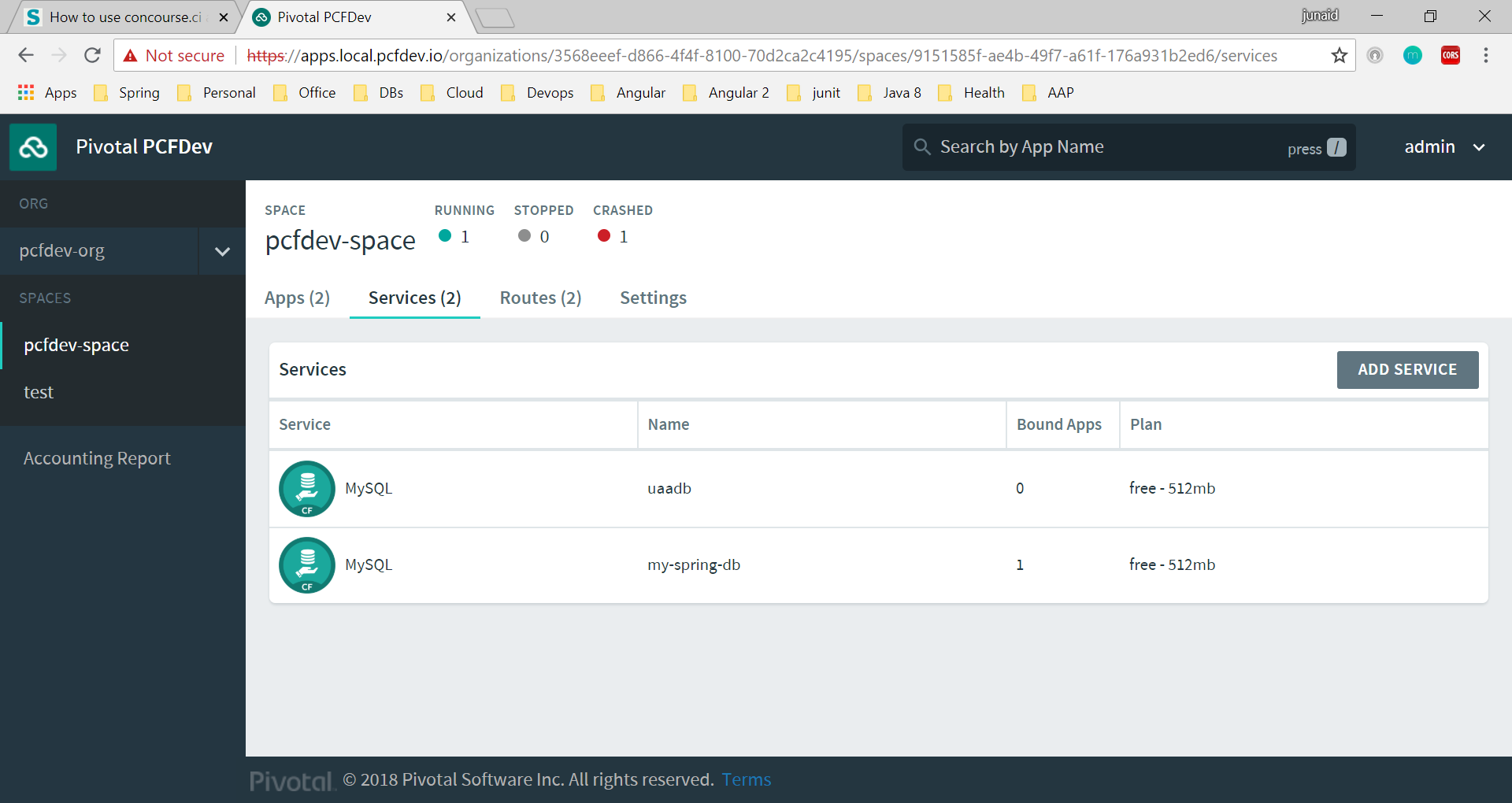


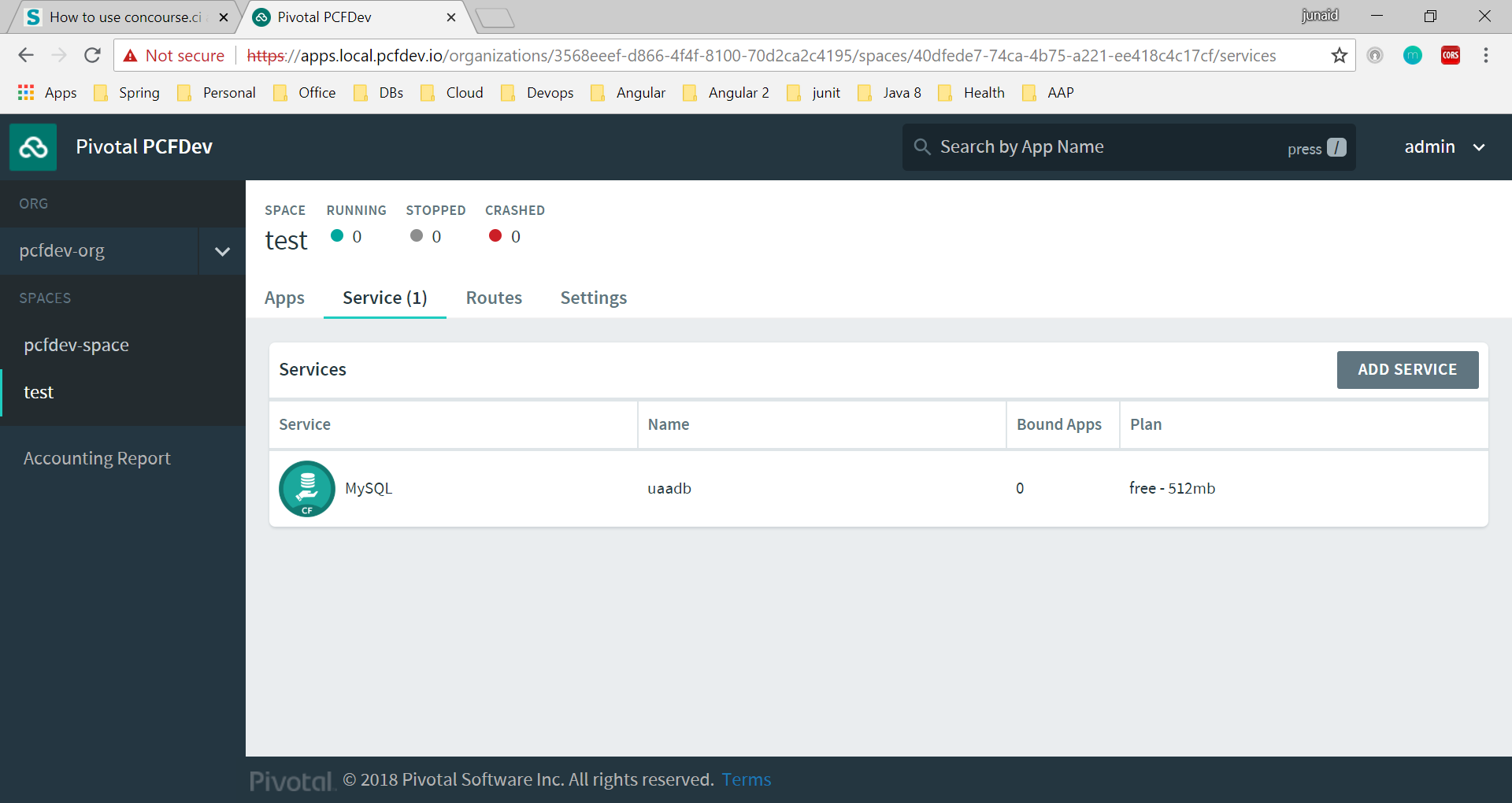




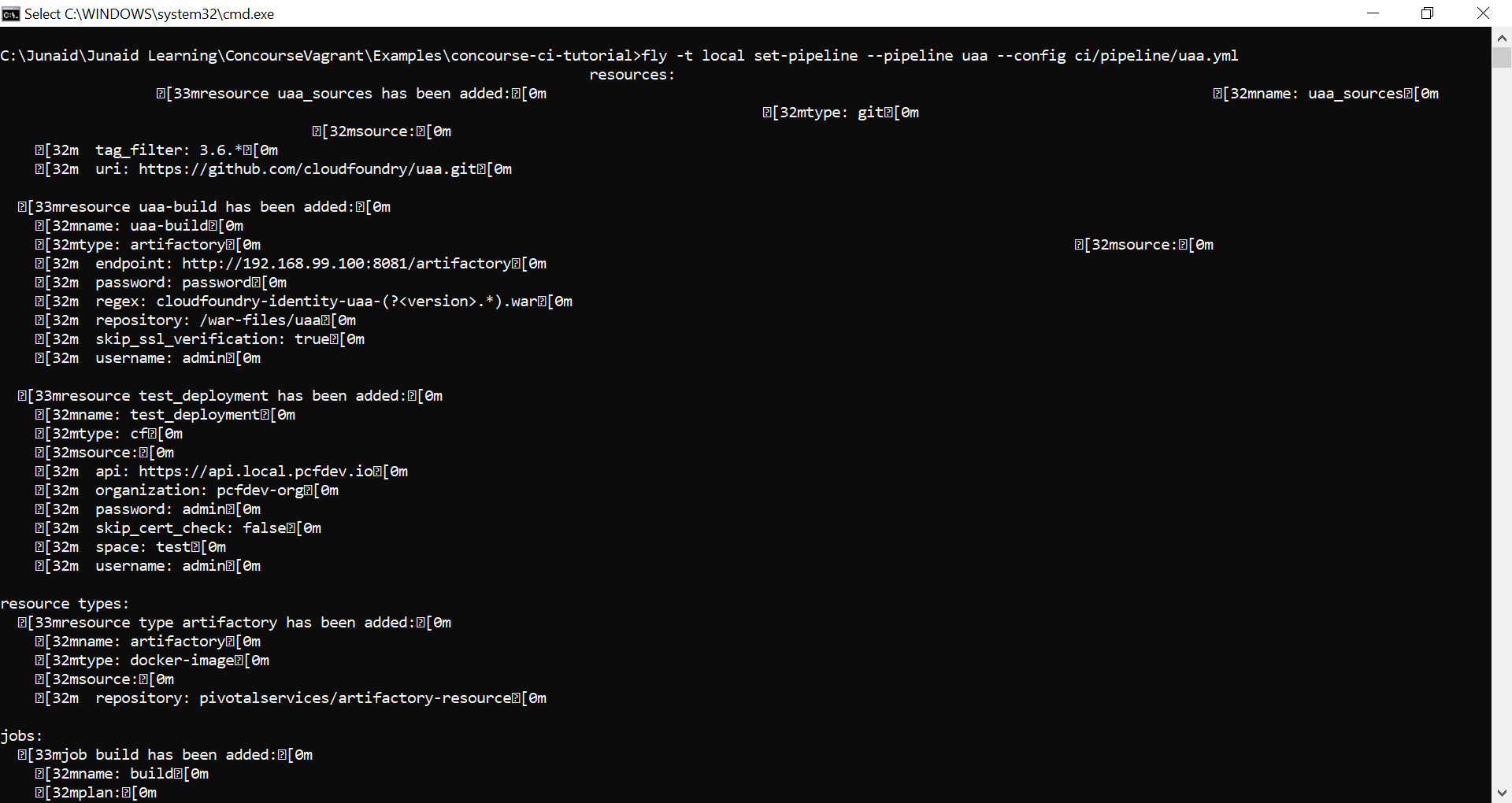
Adding mysql service to dev and test space

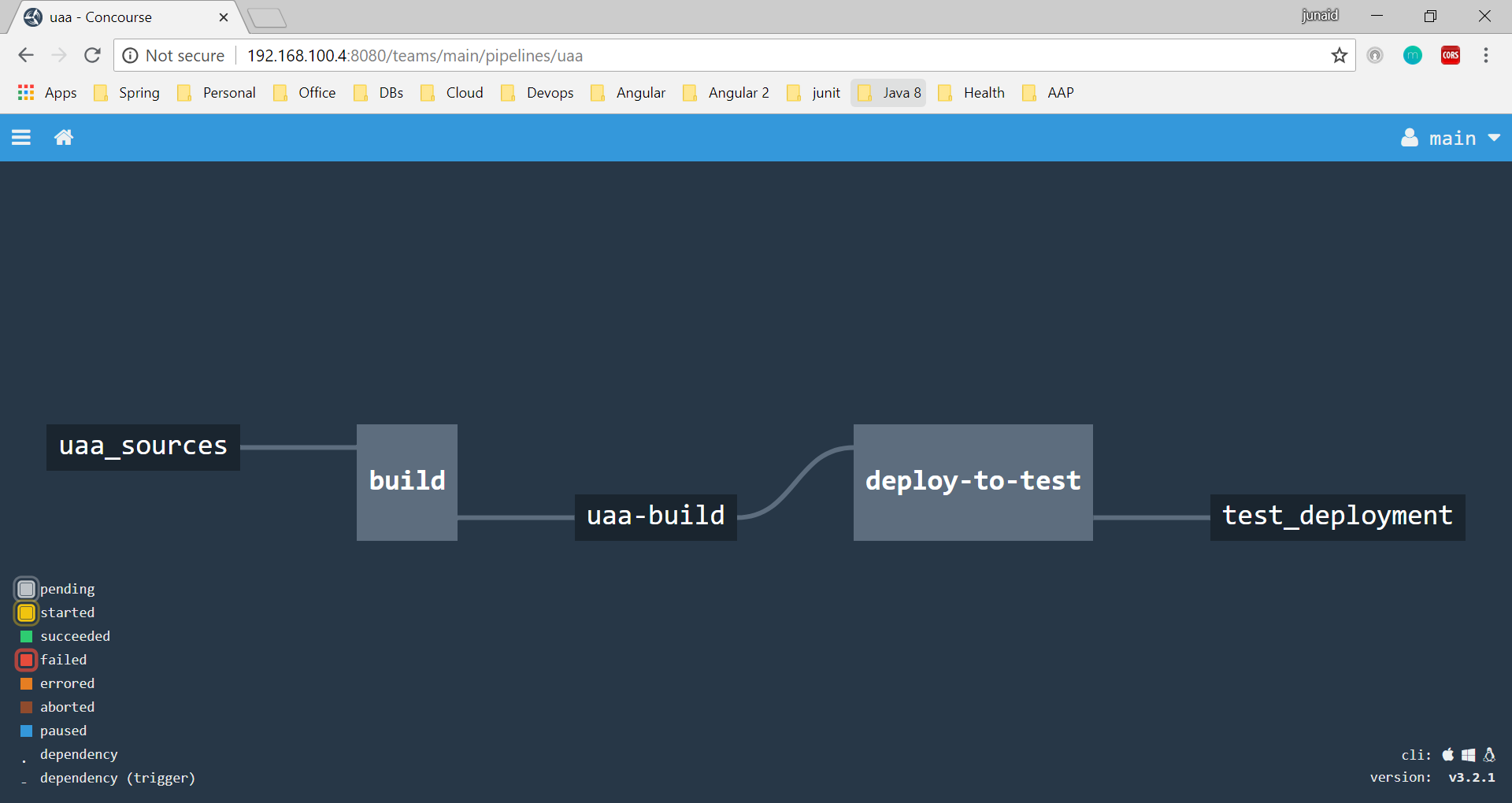


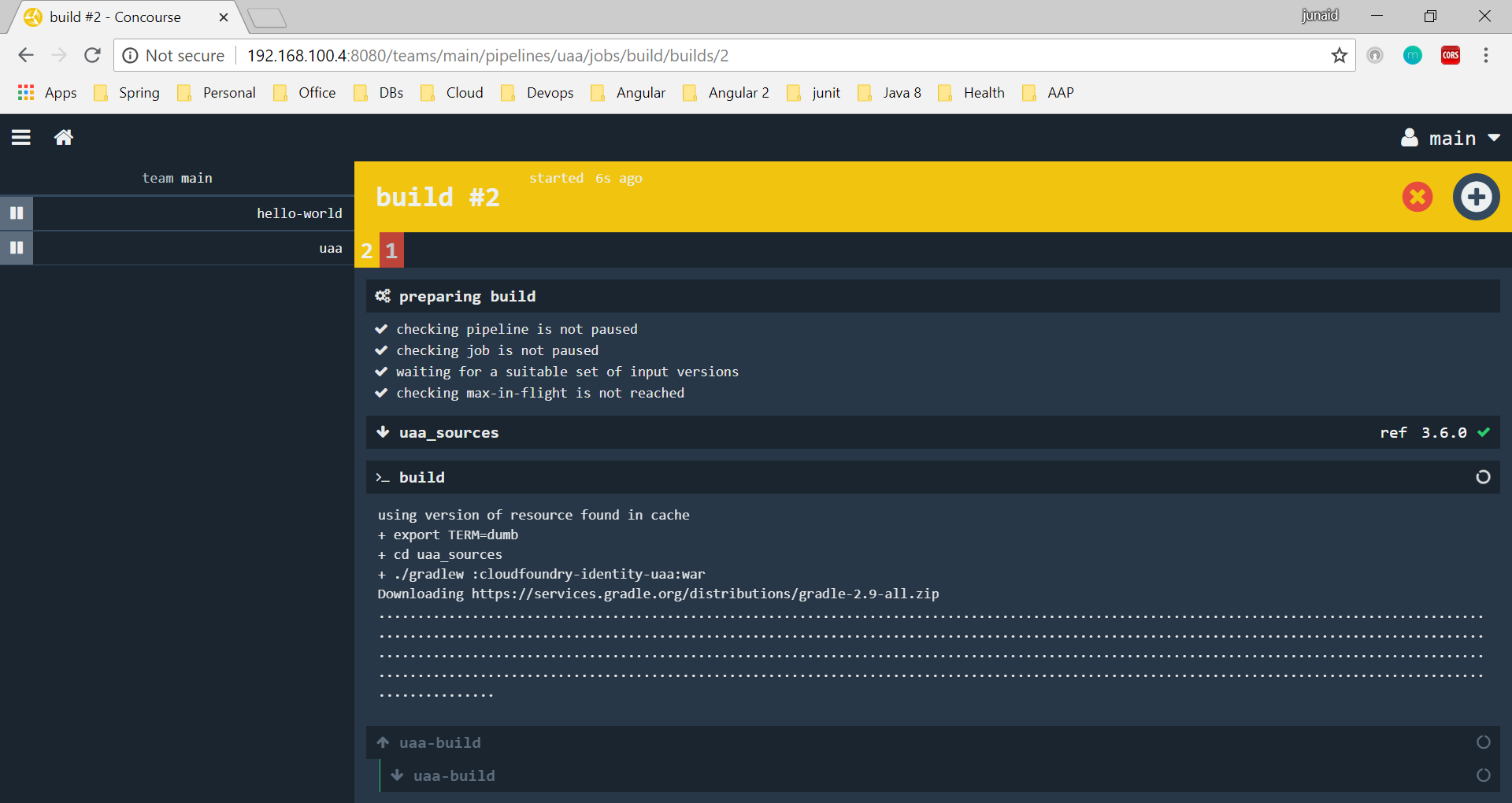


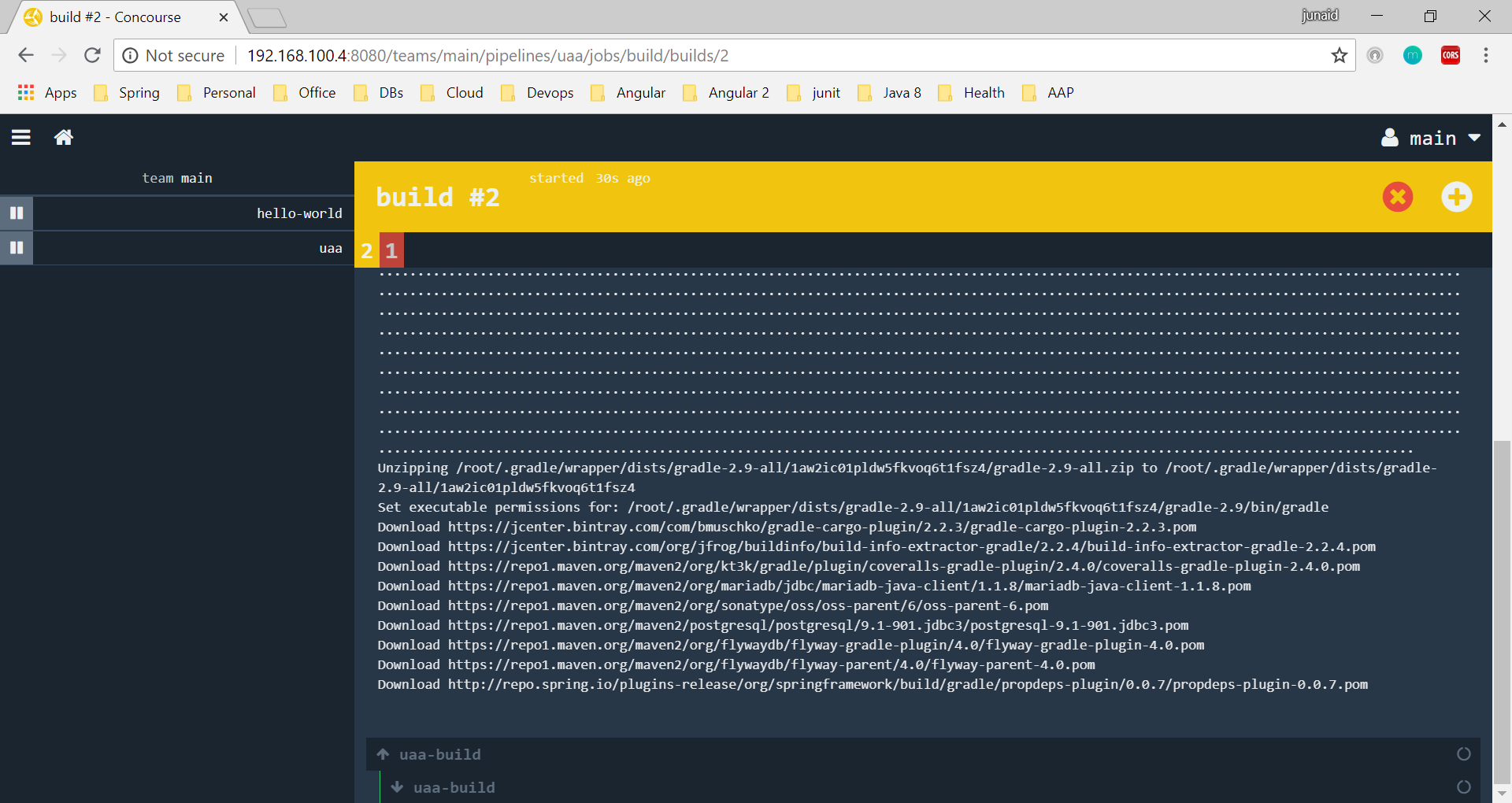


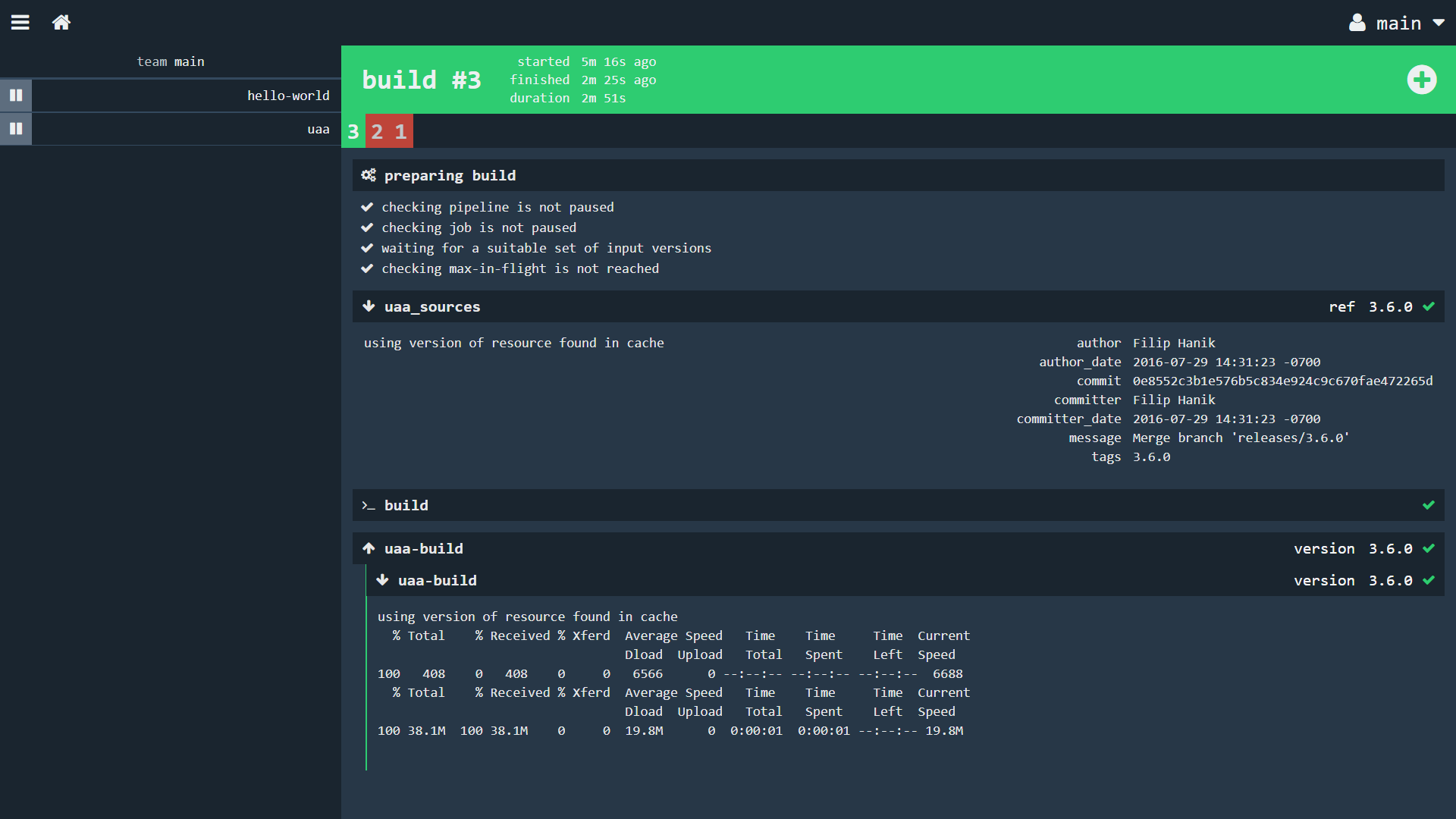
Specify a concourse job to deploy the UAA to the test environment











Specify a concourse job to deploy the UAA to the production environment

