Hello everyone! My name is Sergey. I am from Chernihiv. My current job is a system administrator and computer systems engineer. I am an open person with a desire for self-development and learning.

So, that's why I'm here.

Let me introduce my project:

The main goal of the project, as I understood it, is to make the mini ci / cd process.

This slide shows what I wanted to realize: the developer pushes the code into the repository, some service / tool (for example, Jenkins) takes it from the repository, builds it, tests it, and delivers the result of these steps (artifact) to the next level of the development process (testing , QA environment, production environment). And in each step, we should have notifications about the status of the process.

When I analyzed the structure, I had a lot of questions:

* I have a resource limit (time, money, equipment, experience…), for this reason what tools can I use?
* What location of servers, services, network type should I choose? For example, cloud solutions, VirtualBox, Linux OS or Microsoft Windows OS, MacOS.
* How can Jenkins know about a new commit in the repository (push and pull paths)?
* I need a reproducible and fast creation of infrastructure with backup.
* I need to be notified about the state of the pipeline.
* I need the secure process.

Truthfully (actually), I was a little confused.

So, I decided to deal with this (разобраться с этим) and find solution.

I have a money limit. So, all tools should be open/ free.

I have a time limit so all tools should be user friendly and have a lot of good manual.

I have poor internet connection at home, so I can lose access to the cloud infrastructure at any time. So, I have to use local resources (for example, Virtual Box) and my local network.

If I use my local network without a static white IP or some kind of domain name (for use with DDNS), I can't use WebHooks from other services (GitHUB, DockerHUB). So, I have to use the pull method for Jenkins. We can meet with some problems. Namely, (а именно) jenkins generates some traffic all the time, uses some resource. Regardless, (or in spite of it не смотря на это) in this case, it is the solution.

I need to have some automation to create the infrastructure. So, I decided to use Vagrant. And in the condition of the current project, this is enough. But if the project is larger, I will use Ansible to set up the servers.

For notifications, I chose the Telegram messenger as one of the most popular messenger and, к тому же, it has a lot of bots and good manual.

For more security, I decided to use more ssh connection with passphrases, tokens, additional encryption and a crypto repository to store credentials in Jenkins.

As a result, this is a list of tools that I used in my project:

.........

And this is the structure of my pipeline:

.........

CI Process (Continuous Integration):

The developer submits the code to Git HUB.

Jenkins job sends a request to GIT HUB 1 time per minute. When he sees the changes in the repository, the pipeline starts.

The job starts (запускает – executes) a Docker container with Maven that builds and tests the project.

The result of the CI process is ARTIFACT.

CD process (continuous delivery):

In the current pipeline, delivery starts with building a docker image.

When the image is done, it is sent to Docker Hub.

From Docker Hub, the image can be pushed to a test environment, to a quality assurance environment, to a production / production server.

I made my project in two levels.

First level is a simple java code that prints "HelloWorld!". The test for this is checking the correction of "Hello World!" message. The result of pipeline status is sent to the telegram channel.

And after successfully creation of this Job, I wanted to find more interesting variant of realization of the project. I decided to take the Spring-PetClinic project from github and build it in my pipeline. It was a very good experience.

As a result, I have two different Jenkins files for two projects.

I had my first and main issue when was trying to build and push the artifact to the DockerHub repo. It is due to Maven runs in a docker container without docker inside. And I couldn't start docker building and push inside container. I unterstood that in this case, I can build my own docker image with maven and docker. But as a result, my image had a large size. So, I decided to rewrite the jenkins file in multi-agent style and run the docker build/push commands in another agent.

But everything works.

So, we can see the process of work, notification of various pipeline states, the launch of the Java -jar command , the result.