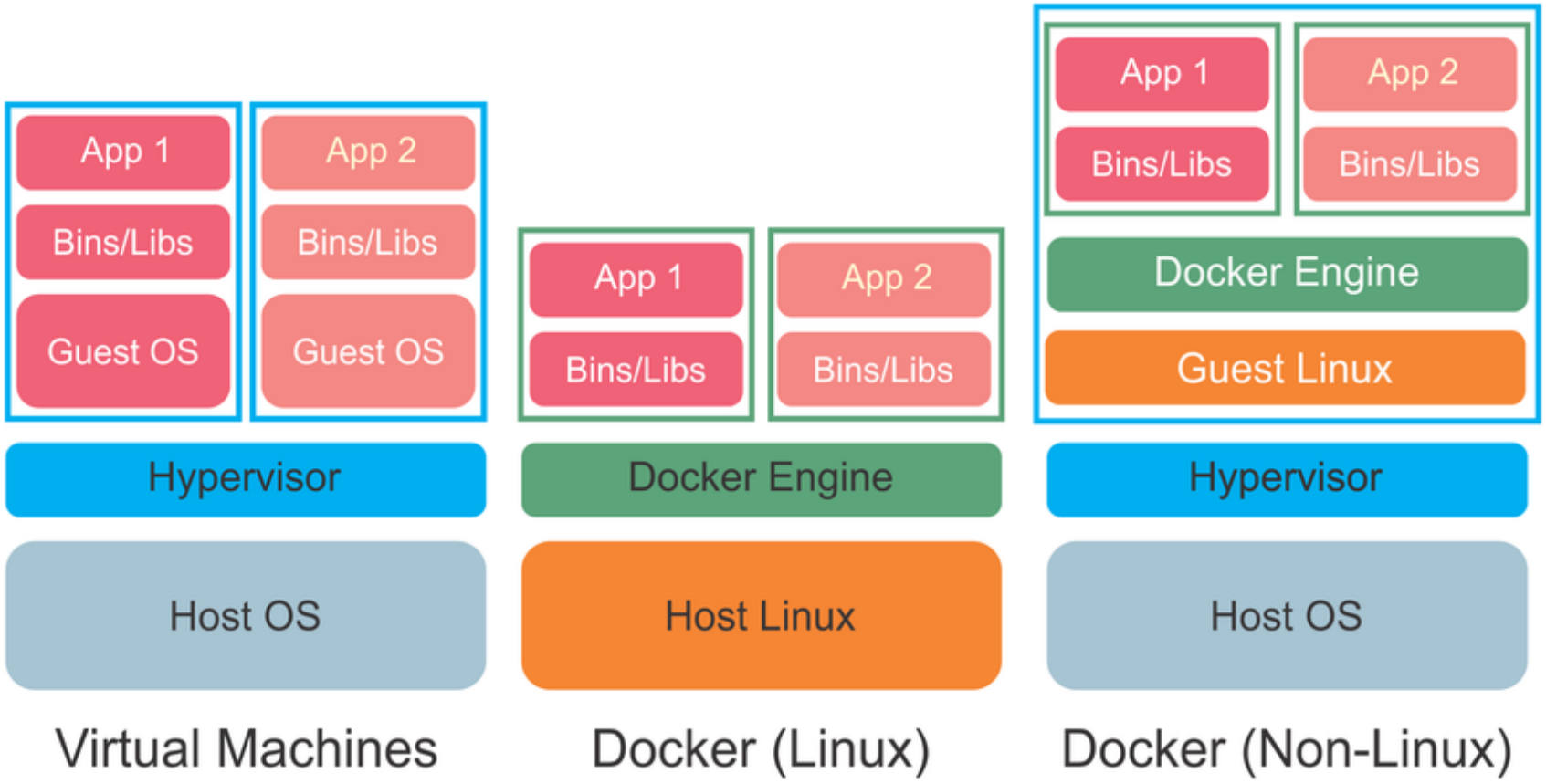
DevOps Project

Deploying web applications in a multi-cloud environment using CI / CD process

Volodymyr Verholyak
EPAM [OnlineUA] Summer 2020

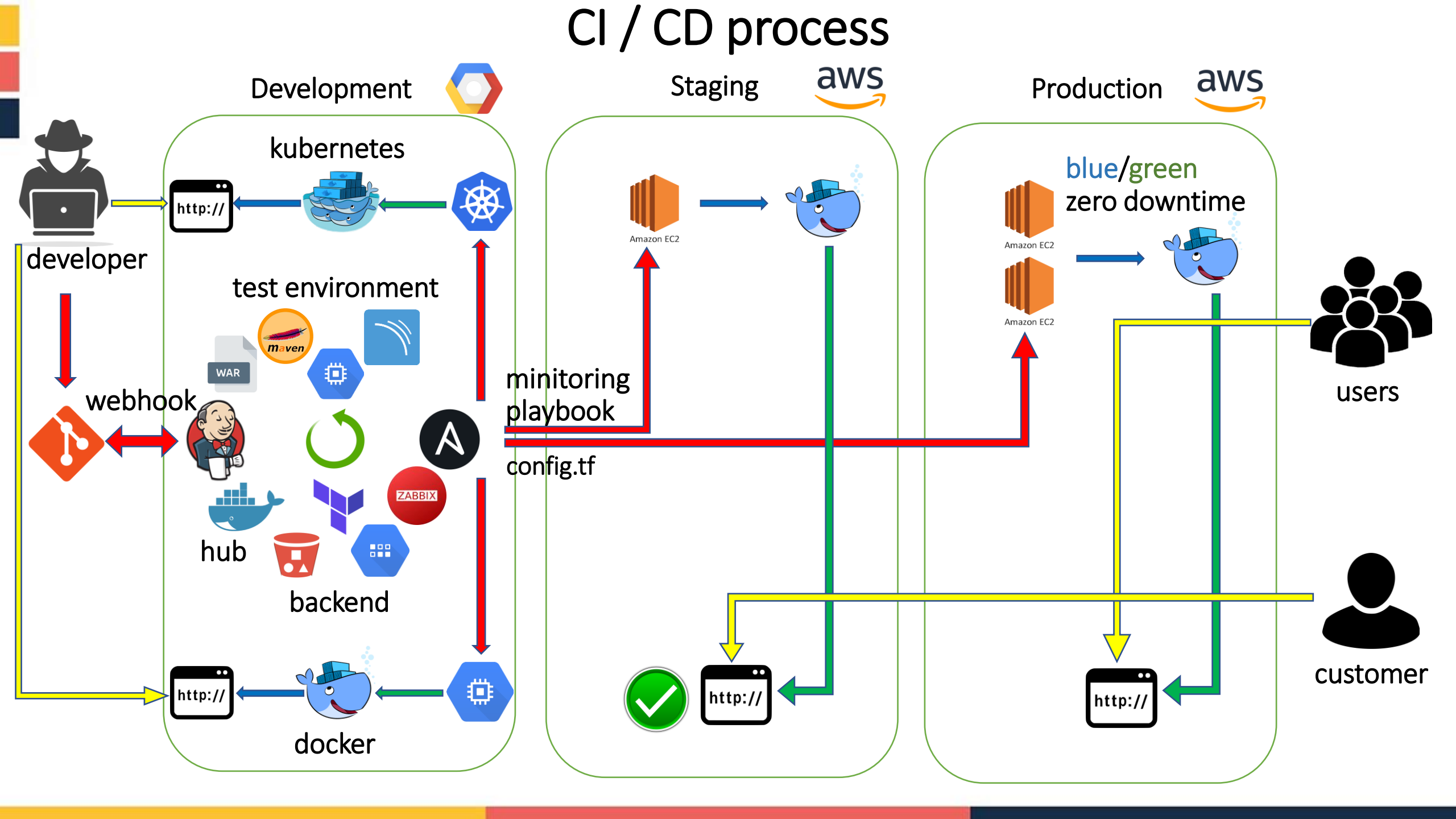
A few words about: Docker

```
FROM tomcat:8.0

LABEL mainteiner Vova_Verholyak

COPY ./webapp.war /usr/local/tomcat/webapps
```

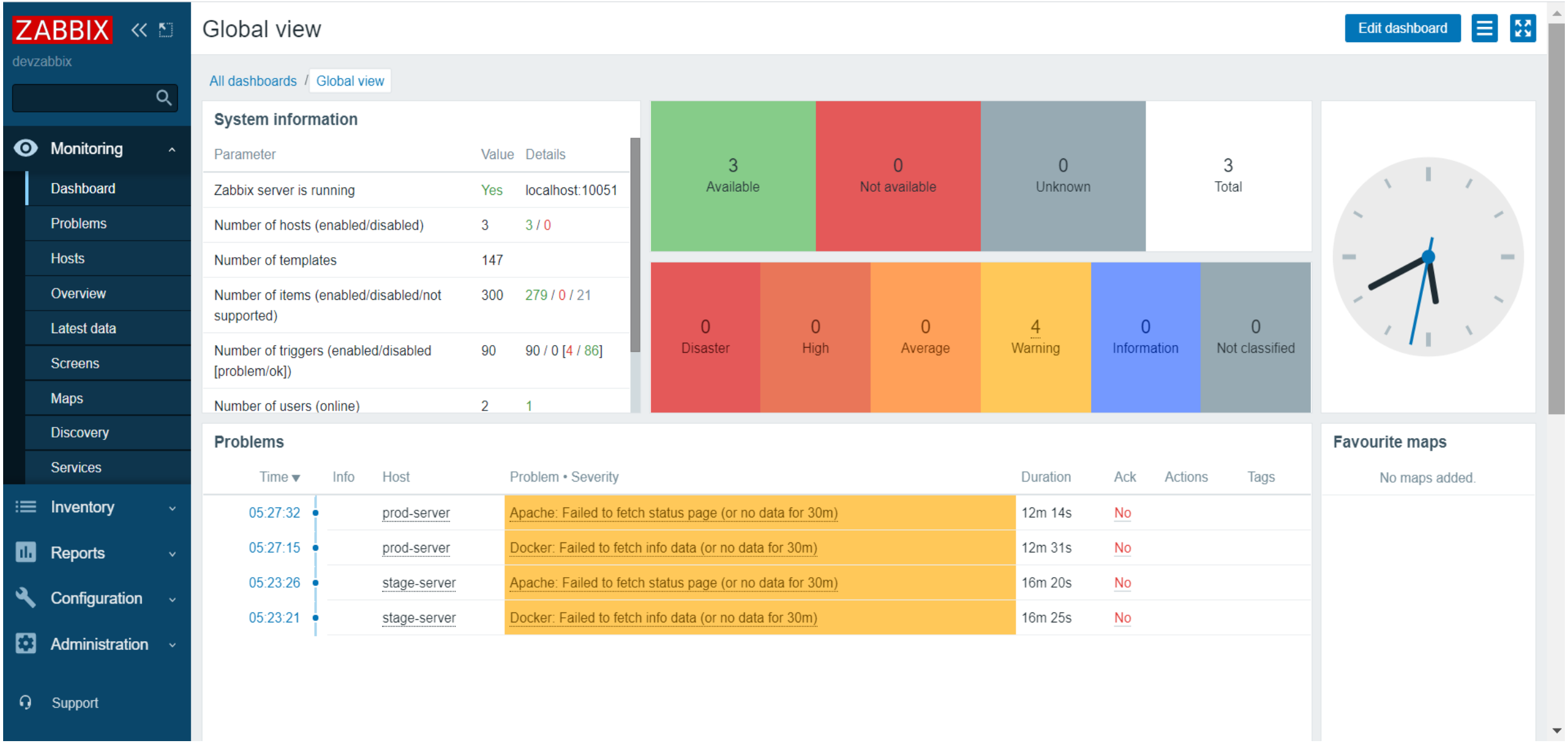


Used tools and services

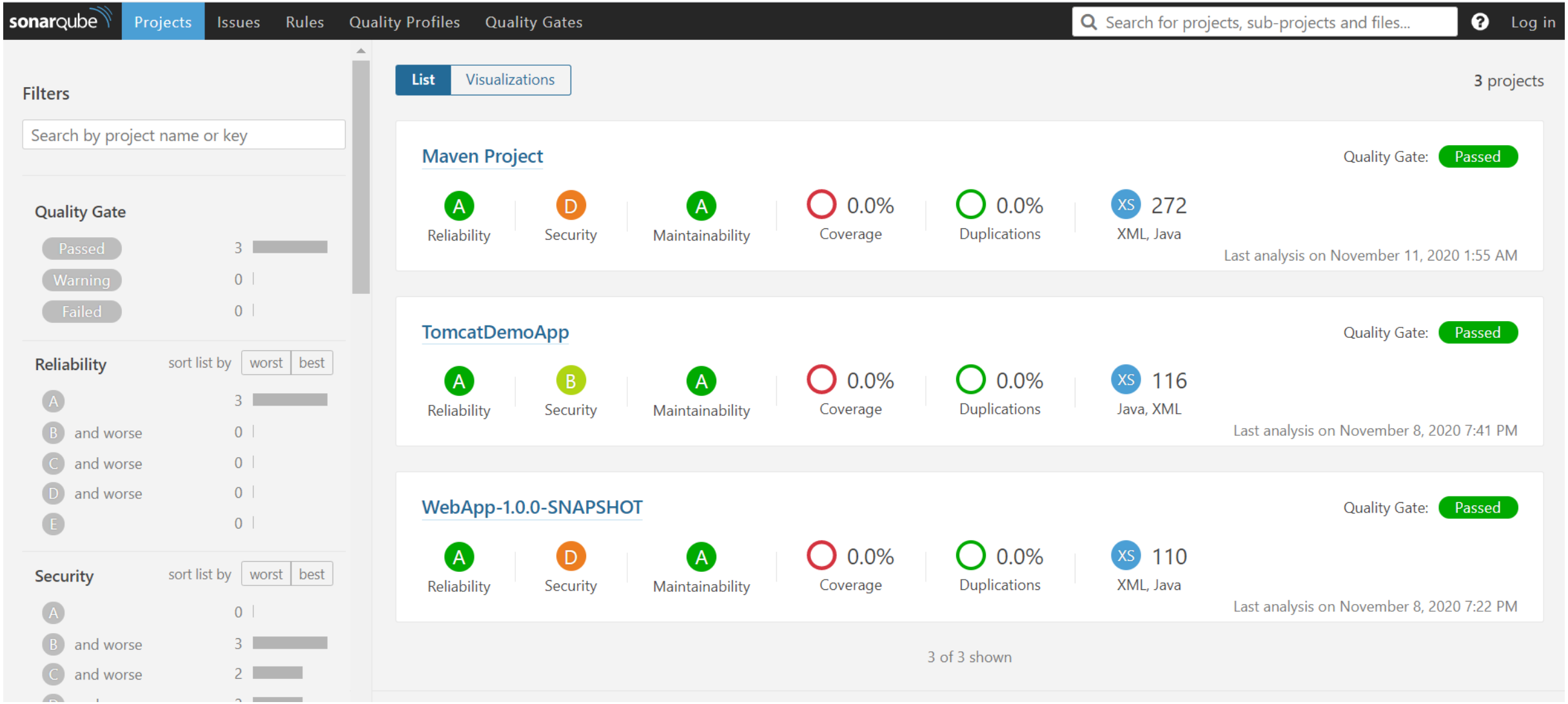
CI / CD process



Zabbix monitoring



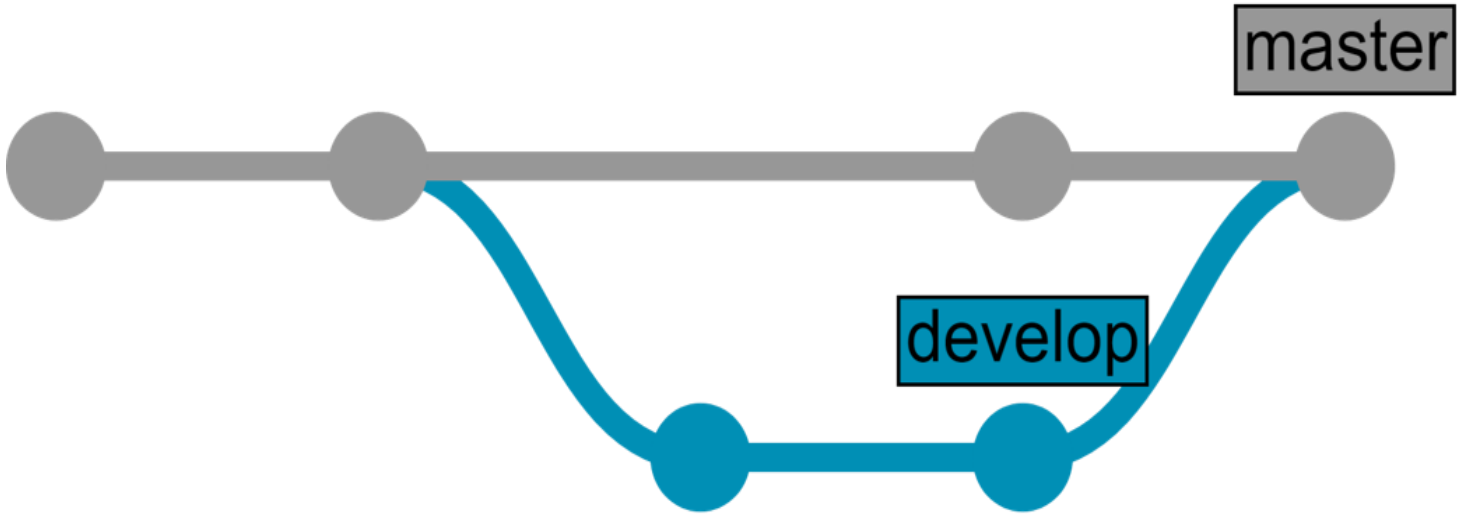
SonarQube test



Step 1. Deploy code or changes

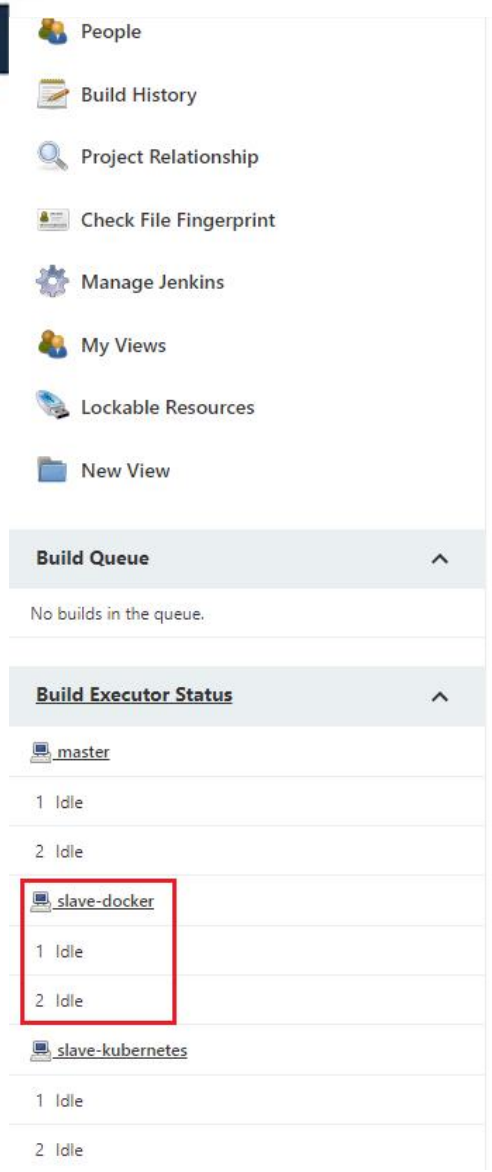
```
TERMINAL   PROBLEMS   OUTPUT   DEBUG CONSOLE

User@DESKTOP-6FBBSGI MINGW64 /c/git_epam/hello-world (master)
$ git checkout tester
Switched to branch 'tester'

User@DESKTOP-6FBBSGI MINGW64 /c/git_epam/hello-world (tester)
$ git add .

User@DESKTOP-6FBBSGI MINGW64 /c/git_epam/hello-world (tester)
$ git commit -m "added changes by tester"
[tester 41a57d7] added changes by tester
 1 file changed, 2 insertions(+), 1 deletion(-)

User@DESKTOP-6FBBSGI MINGW64 /c/git_epam/hello-world (tester)
$ git push --all
Enumerating objects: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 570 bytes | 142.00 KiB/s, done.
Total 7 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To github.com:verholyak/hello-world.git
   933997b..41a57d7  tester -> tester

User@DESKTOP-6FBBSGI MINGW64 /c/git_epam/hello-world (tester)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

User@DESKTOP-6FBBSGI MINGW64 /c/git_epam/hello-world (master)
$ git merge tester
Merge made by the 'recursive' strategy.
 webapp/src/main/webapp/index.jsp | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)

User@DESKTOP-6FBBSGI MINGW64 /c/git_epam/hello-world (master)
$ git add .

User@DESKTOP-6FBBSGI MINGW64 /c/git_epam/hello-world (master)
$ git commit -m "web v10"
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

User@DESKTOP-6FBBSGI MINGW64 /c/git_epam/hello-world (master)
$ git push --all
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 279 bytes | 279.00 KiB/s, done.
Total 2 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To github.com:verholyak/hello-world.git
   eda0147..1bb40ad  master -> master

User@DESKTOP-6FBBSGI MINGW64 /c/git_epam/hello-world (master)
$
```

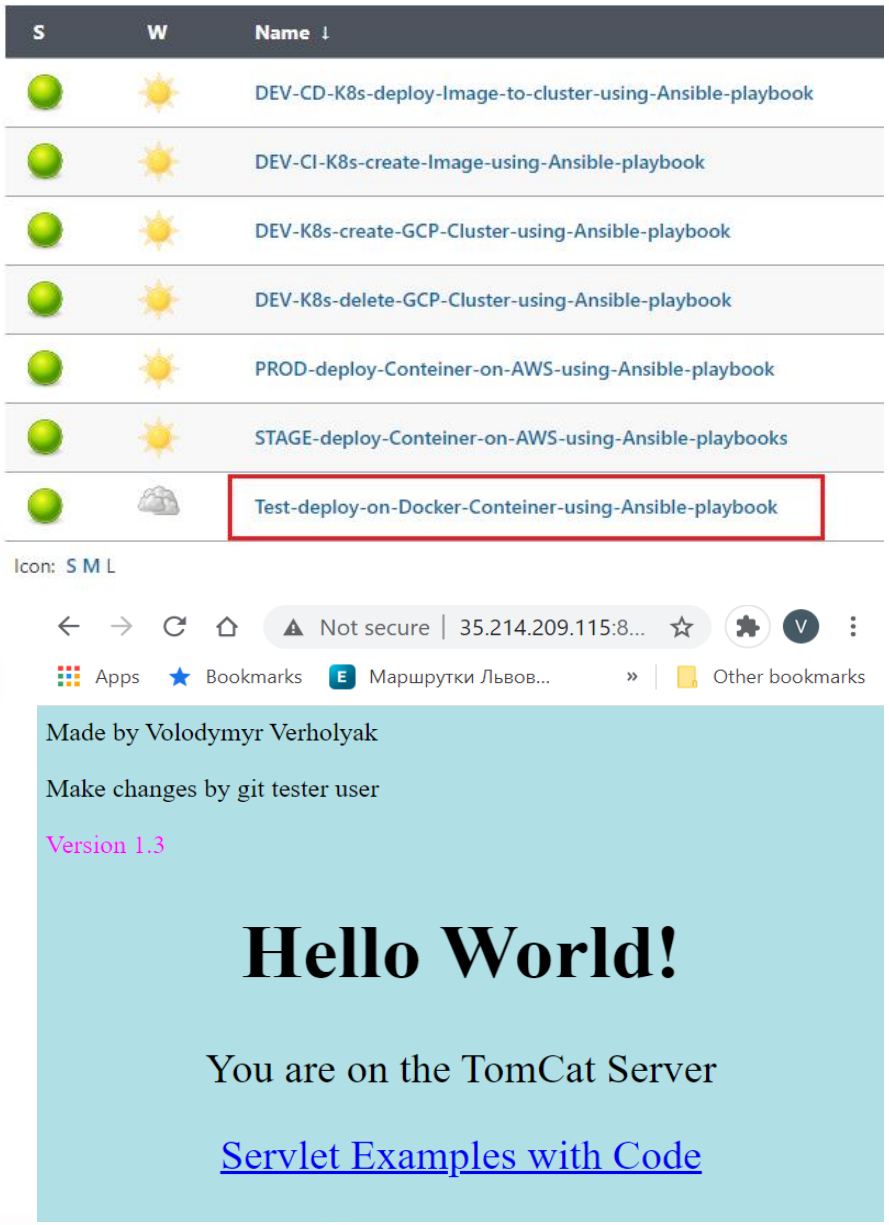
Developer wants to make some changes to the code and he wants to see the changes → let's start !!!



Webhook to Jenkins →

Step 2. Deploy to Docker Dev-Server using Ansible playbook



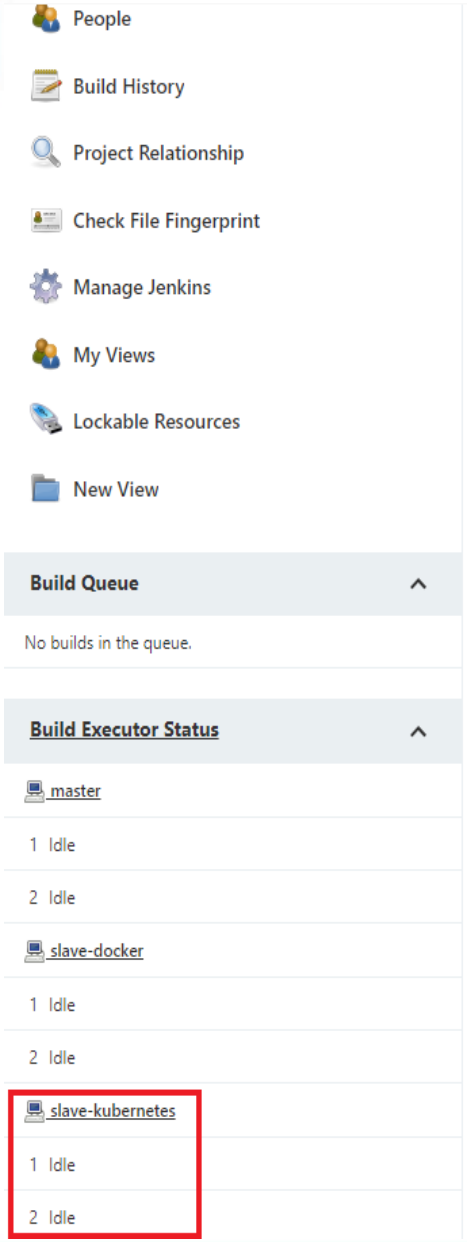


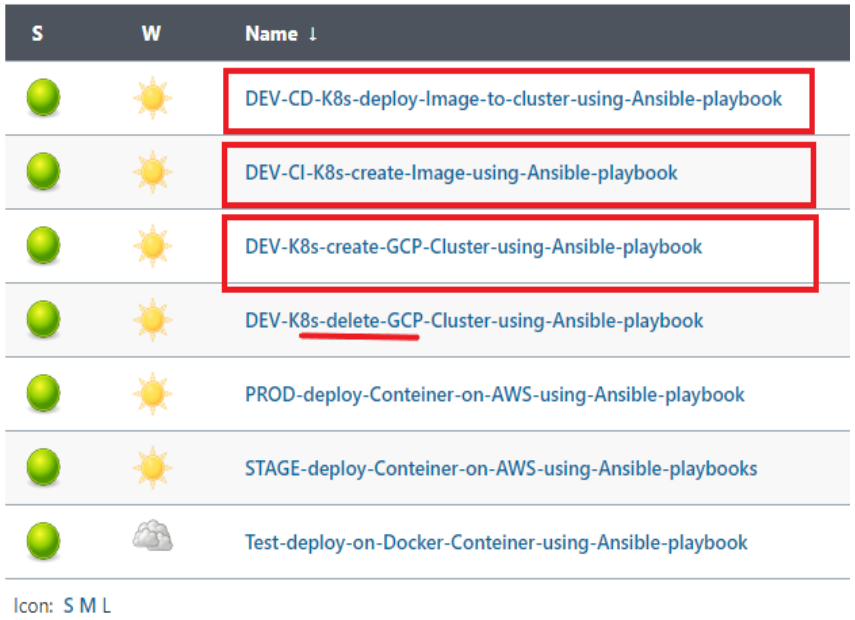
what I used here:

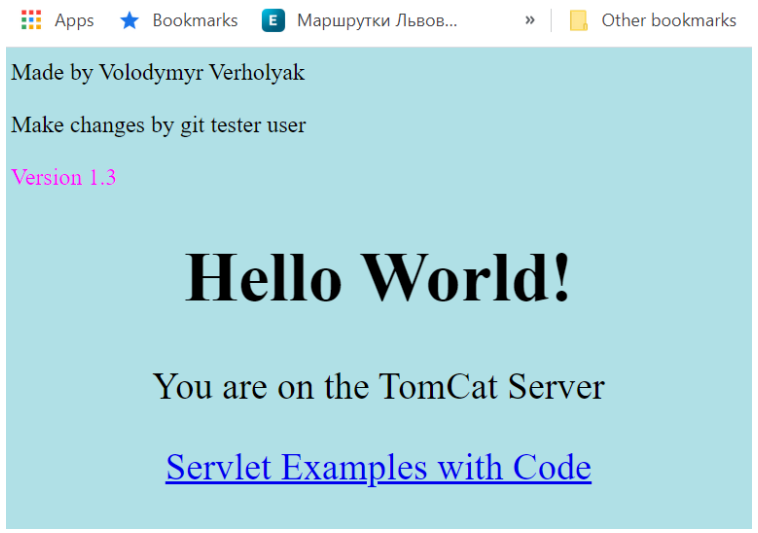
- GitHub webhook and services ngrok
- Maven, SonarQube test
- Create artifact “webapp.war” file
- Ansible playbook, create image and push to DockerHub repository
- Ansible playbook, deploy image from DockerHub repository to Dev-Server
- Use GCP cloud provider
- http://35.214.209.115:8081/webapp/

Deploy to Kubernetes →

Step 3. Deploy changes to Kubernetes from Jenkins job (manual)





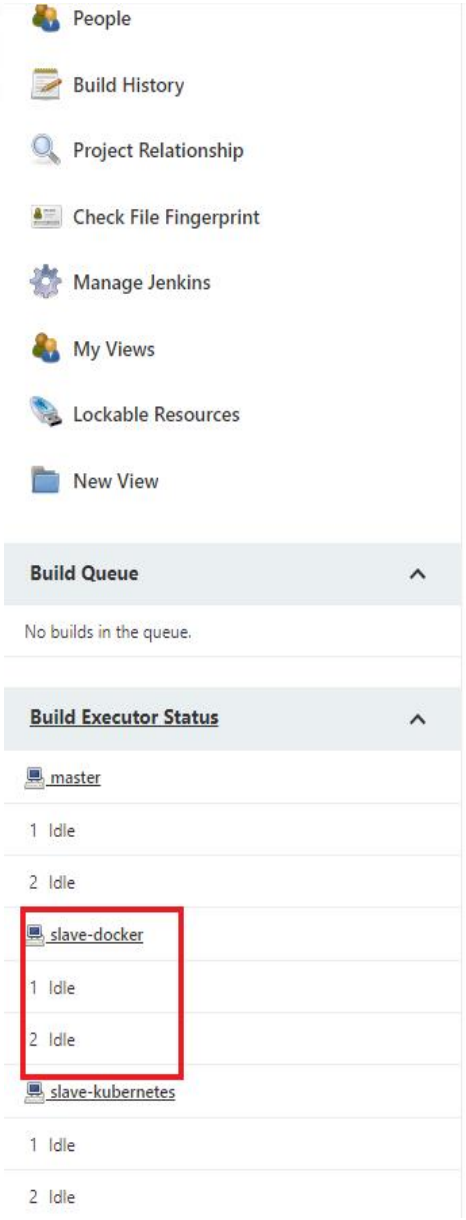


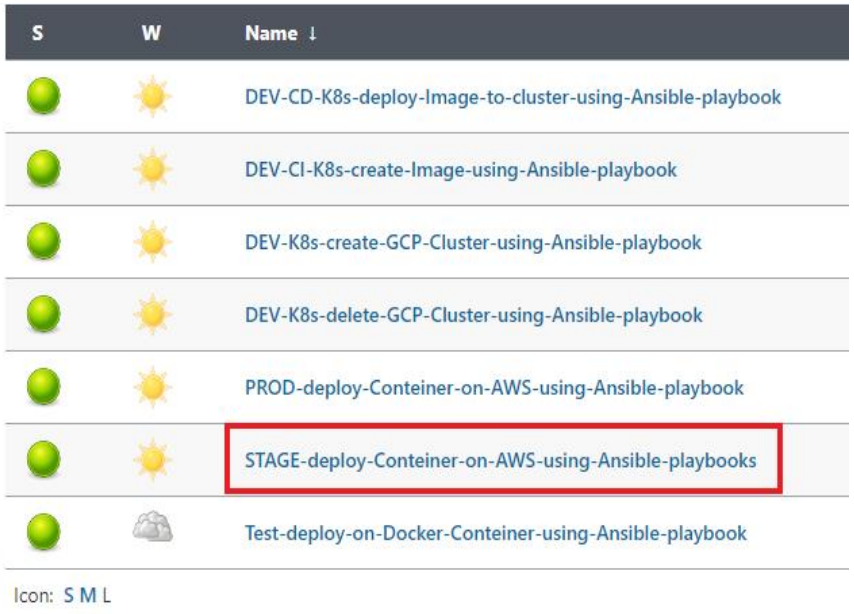
what I used here:

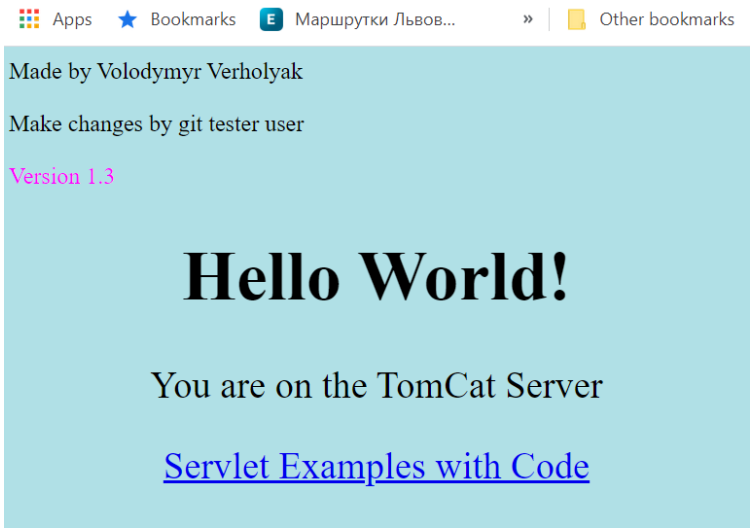
- Maven, SonarQube test
- Create artifact “webapp.war” file
- Ansible playbook, create Kubernetes Cluster, using parameters
- Ansible playbook, create image and push to DockerHub repository
- Ansible playbook, deploy image from DockerHub repository to Kubernetes Cluster using yaml config
- Use GCP cloud provider
- http://35.204.36.248:8081/webapp/

Deploy to STAG using Jenkins →

Step 3. Deploy to STAGE WebServer using Jenkins job (manual)





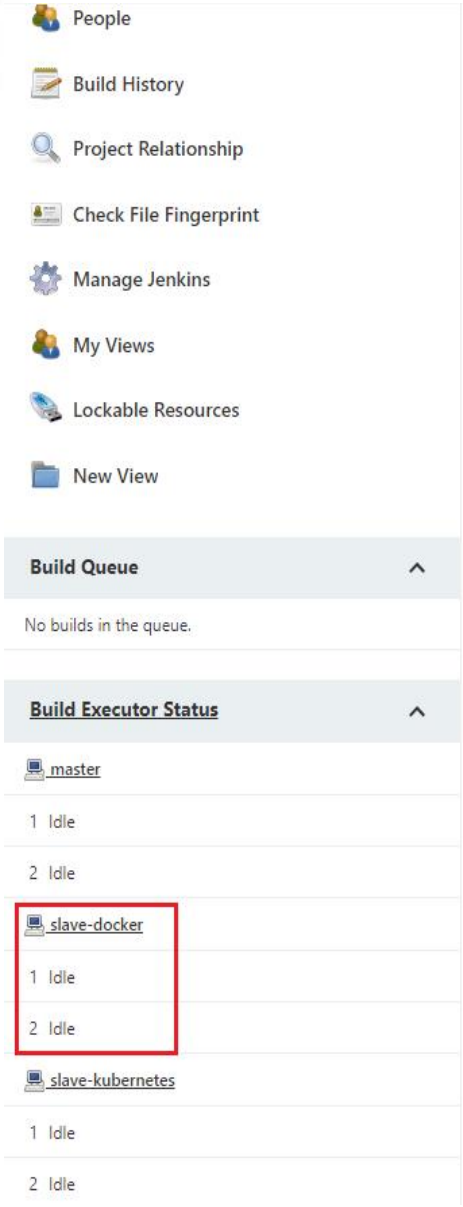


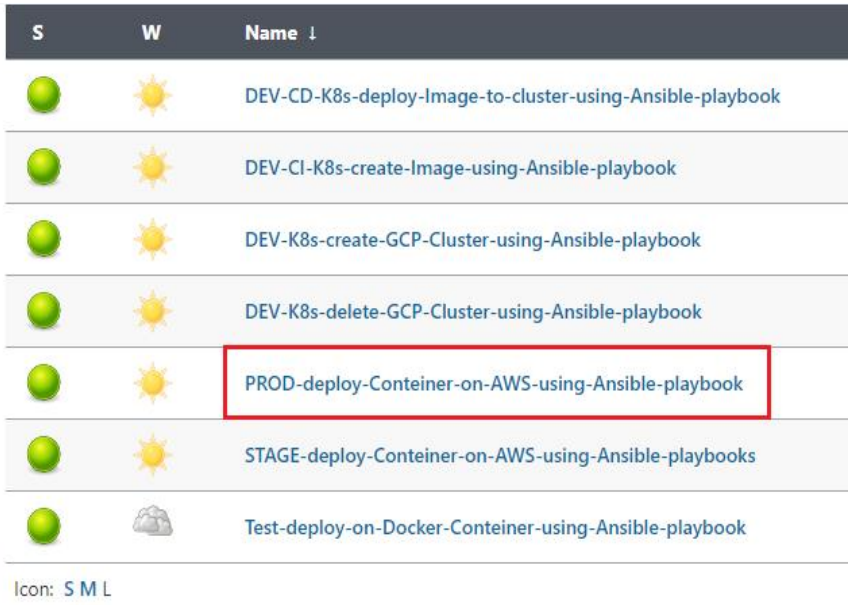
what I used here:

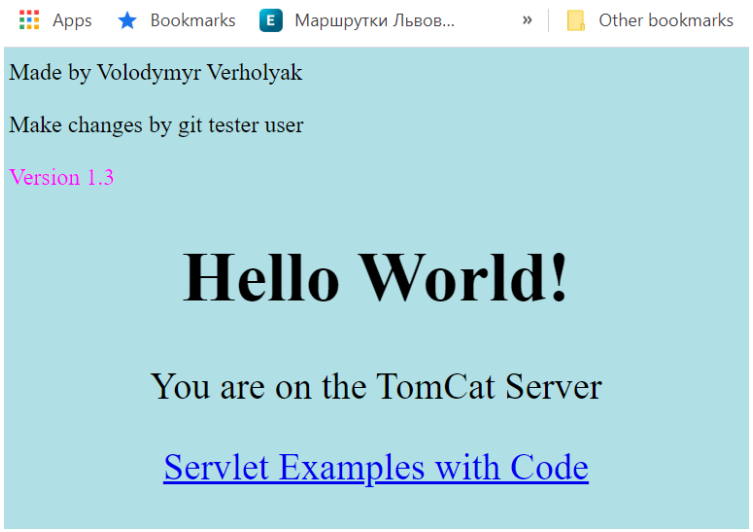
- Maven, SonarQube test
- Create artifact “webapp.war” file
- Ansible playbook, create image and push to DockerHub repository
- Ansible playbook, deploy image from DockerHub repository to STAGE-Server
- Use AWS cloud provider
- http://18.198.63.214:8081/webapp/

Deploy to PROD using Jenkins →

Step 4. Deploy to PROD WebServer using Jenkins job (manual)







what I used here:

- Maven, SonarQube test
- Create artifact “webapp.war” file
- Ansible playbook, deploy image from DockerHub repository to STAGE-Server
- Use AWS cloud provider
- http://18.159.26.114:8081/webapp/

Deploy to DEV using Jenkins →

developer, customer and users are happy because their service works well :)

Infrastructure in pictures

Google Cloud Platform

A Kubernetes cluster is a managed group of VM instances for running containerized applications. Learn more

Filter by label or name

Name	Location	Cluster size	Total cores	Total memory	Notifications	Labels	
demo-k8s	europe-west4-a	4	8 vCPUs	16.00 GB			Connect

Name	Zone	Internal IP	External IP	Connect
gke-demo-k8s-default-pool-7201793d-0bvm	europe-west4-a	10.164.15.205 (nic0)	34.90.0.89	SSH
gke-demo-k8s-default-pool-7201793d-wx6j	europe-west4-a	10.164.15.204 (nic0)	34.90.228.124	SSH
gke-demo-k8s-default-pool-7201793d-xhx5	europe-west4-a	10.164.15.203 (nic0)	35.204.129.127	SSH
gke-demo-k8s-default-pool-7201793d-zwg3	europe-west4-a	10.164.15.202 (nic0)	35.204.36.248	SSH
server-docker-aws	europe-west4-a	10.164.0.10 (nic0)	35.214.209.115	SSH
server-sonarqube	europe-west4-a	10.164.0.61 (nic0)	34.91.174.170	SSH

amazon web services

Instances (2) Info

Connect

Filter instances

Name	Instance ID	Instance state	Instance type	Status check
PROD WebSe...	i-0f7f5b3b0412cc2c3	Running	t2.micro	2/2 checks ...
STAG WebSe...	i-07776e49c9cfd66d6	Running	t2.micro	2/2 checks ...

Security Groups (3) Info

Filter security groups

Name	Security group ID	Security group name	VPC ID
Dynamic Security G...	sg-04c0c85457d5aef00	PROD Dynamic SG	vpc-444f8d2e
Web Server Securit...	sg-0d182f37301f0d1d4	WebServer SG Stag	vpc-444f8d2e
my-own-sg	sg-1889667f	default	vpc-444f8d2e

Create Load Balancer | Actions

Filter by tags and attributes or search by keyword

Name	DNS name	State	VPC ID
WebServer-HA-ELB	WebServer-HA-ELB-646341...		vpc-444f8d2e

Elastic IP addresses (2)

Filter Elastic IP addresses

Name	Allocated IPv4 add...	Type
PROD WebServer IP	18.159.26.114	Public IP
STAGE WebServer IP	18.198.63.214	Public IP

Auto Scaling groups (1)

Search your Auto Scaling groups

Name	Launch templ...	Instances	Status	Desired capacity
ASG-WebServer-H	WebServer-Highl...	1	-	1

Terraform: infrastructure as code

Preview README.MD | index.jsp | main.tf

terraform > project-epam > prod > main.tf

```
13
14  provider "aws" {
15    region = var.region     // variables
16  }
17
18  data "aws_availability_zones" "available" {}
19  data "aws_ami" "latest_amazon_linux" {
20    owners      = ["amazon"]
21    most_recent = true
22    filter {
23      name   = "name"
24      values = ["amzn2-ami-hvm-*-x86_64-gp2"]
25    }
26  }
27
28  #---------------------------------------------------------------
29  resource "aws_security_group" "web" {
30    name = "PROD Dynamic SG"
31
32    dynamic "ingress" {
33      for_each = var.allow_ports        // variables
34      content {
35        from_port   = ingress.value
36        to_port     = ingress.value
37        protocol    = "tcp"
38        cidr_blocks = ["0.0.0.0/0"]
39      }
40    }
41    egress {
42      from_port   = 0
43      to_port     = 0
44      protocol    = "-1"
45      cidr_blocks = ["0.0.0.0/0"]
46    }
47
48    tags = {
49      Name  = "Dynamic Security Group"
50      Owner = "Vova Verholyak"
51    }
52  }
53
```

backend.tf | user_data.sh

terraform > project-epam > prod > user_data.sh

```
1   #!/bin/bash
2   yum -y update
3   yum -y install httpd
4
5   myip=`curl http://169.254.169.254/latest/meta-data/local-ipv4`
6
7   cat <<EOF > /var/www/html/index.html
8   <html>
9   <head>
10  <title>Simple Web Page</title>
11  </head>
12  <body>
13  <font color="black">Made by Vova Verholyak<br><br>
14  <font color="black">Name Server: <font color="red">Production<br>
15  <font color="black">PrivateIP: <font color="red">$myip<br>
16  <font color="magenta">Version 3.1<br>
17  <font color="black">
18  <h1 align="center">Hello World!</h1>
19  <p align="center">This is my web page</p>
20  </body>
21  </html>
22  EOF
23
24  sudo service httpd start
25  chkconfig httpd on
26
27  yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rp
28  yum install stress -y
29
30  sudo amazon-linux-extras install docker -y
31  sudo service docker start
32  sudo useradd -u 12345 -G docker,systemd-journal,wheel,adm  -d /home/ansadmin -s /bin
33  sudo su root
34  echo "ansadmin ALL=(ALL) NOPASSWD: ALL" >> /etc/sudoers
35  sudo su ansadmin
36  docker run -d --name play-devops-container -p 8081:8080 verholyak/play-devops-image:
37  mkdir /home/ansadmin/.ssh && ssh-keygen -t rsa -f /home/ansadmin/.ssh/id_rsa -q -P "
```

This is only a part of the code, you can see the whole project source code in my private repository by following the link: https://github.com/verholyak/Epam_External_Project

Ansible playbook

ansadmin@vova-PC: /opt/dev_env/kubernetes

```
ansadmin@vova-PC:/opt/dev_env/kubernetes$ cat playbook-create-devops-image.yml
---
- hosts: local_ansible
#  become: true

  tasks:
  - name: create docker image using war file
    command: docker build -t play-devops-image:latest .
    args:
      chdir: /opt/dev_env/kubernetes

  - name: create tag to image
    command: docker tag play-devops-image verholyak/play-devops-image

  - name: push image to dockerhub
    command: docker push verholyak/play-devops-image

  - name: remove docker image from ansible server
    command: docker rmi play-devops-image:latest verholyak/play-devops-image
    ignore_errors: yes
ansadmin@vova-PC:/opt/dev_env/kubernetes$ cat playbook-create-devops-project.yml
---
- hosts: dev_docker_server
  become: true

  tasks:

  - name: stop current running container
    command: docker stop play-devops-container
    ignore_errors: yes

  - name: remove stopped container
    command: docker rm play-devops-container
    ignore_errors: yes

  - name: remove docker image
    command: docker rmi verholyak/play-devops-image:latest
    ignore_errors: yes

  - name: push docker image from dockerhub
    command: docker pull verholyak/play-devops-image:latest

  - name: create container using play-devops-image
    command: docker run -d --name play-devops-container -p 8081:8080 verholyak/play-devops-image:latest
ansadmin@vova-PC:/opt/dev_env/kubernetes$
```

ansadmin@vova-PC: /opt/dev_env/kubernetes

```
ansadmin@vova-PC:/opt/dev_env/kubernetes$ cat playbook-k8s-gcp-config-deploy.yaml
---
- hosts: local_ansible

  tasks:
  - name: Create Kubernetes Cluster in GCP Cloud
    command: kubectl apply -f /opt/dev_env/kubernetes/k8s-gcp-config-deploy.yaml
    ignore_errors: yes
ansadmin@vova-PC:/opt/dev_env/kubernetes$ cat playbook-k8s-gcp-create-cluster.yaml
---
- hosts: local_ansible

  tasks:
  - name: Create Kubernetes Cluster in GCP Cloud
    command: gcloud container clusters create demo-k8s --num-nodes=4 --machine-type=custom-2-4096
    ignore_errors: yes
ansadmin@vova-PC:/opt/dev_env/kubernetes$ cat playbook-k8s-gcp-config-deploy.yaml
---
- hosts: local_ansible

  tasks:
  - name: Create Kubernetes Cluster in GCP Cloud
    command: kubectl apply -f /opt/dev_env/kubernetes/k8s-gcp-config-deploy.yaml
    ignore_errors: yes
ansadmin@vova-PC:/opt/dev_env/kubernetes$
```

DEMO

Question ?