

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

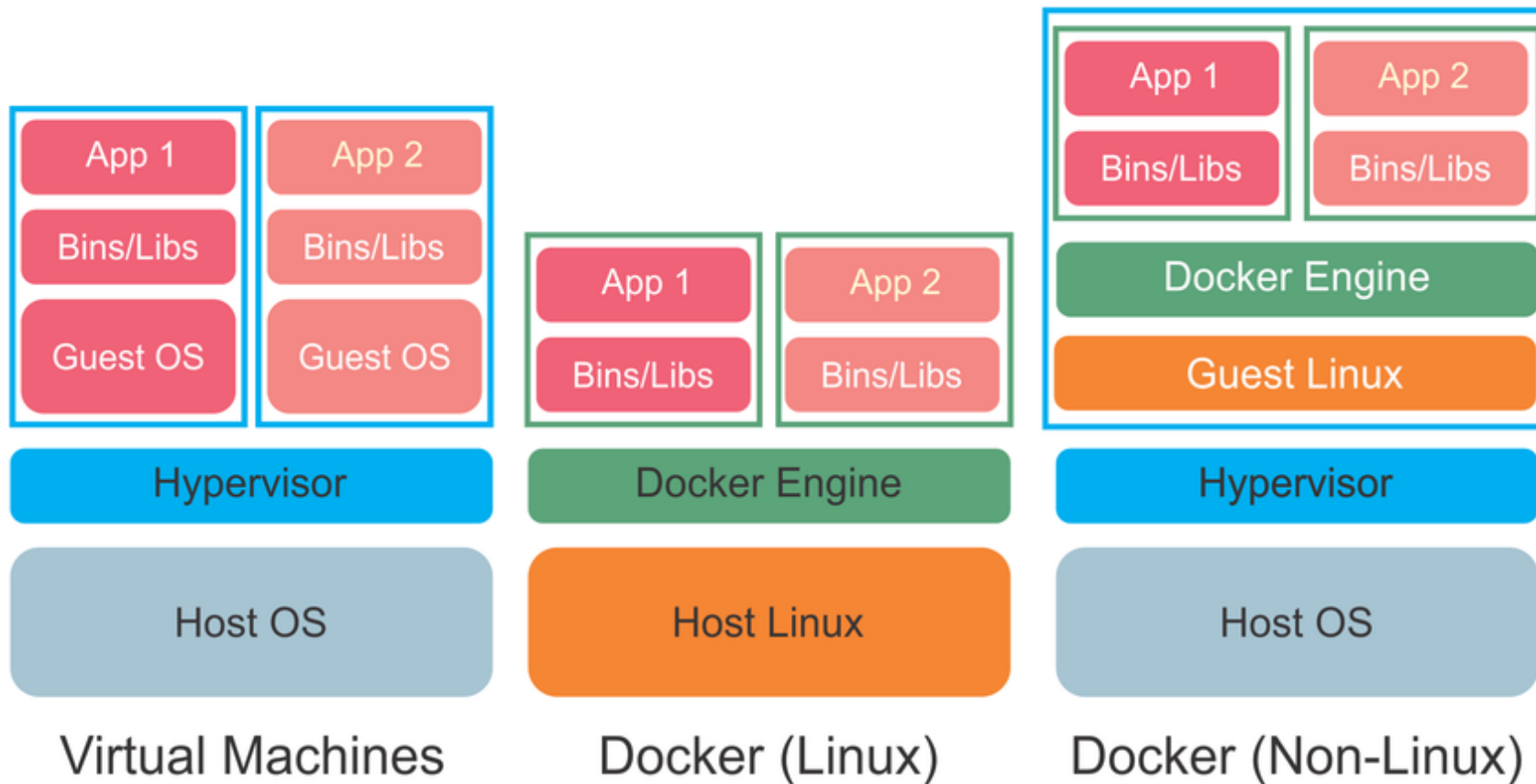
Volodymyr Verholyak  
EPAM [OnlineUA] Summer 2020

# A few words about: Docker

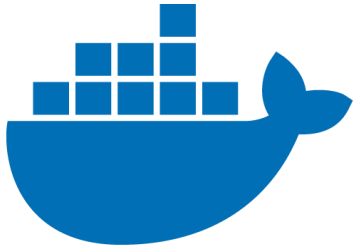
```
FROM tomcat:8.0

LABEL maintainer Vova_Verholyak

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



# Used tools and services



sonarqube



docker

ANSIBLE

ZABBIX

Google Cloud Platform



HashiCorp

Terraform

HTML



Apache  
Tomcat



Jenkins

ngrok



kubernetes



ubuntu



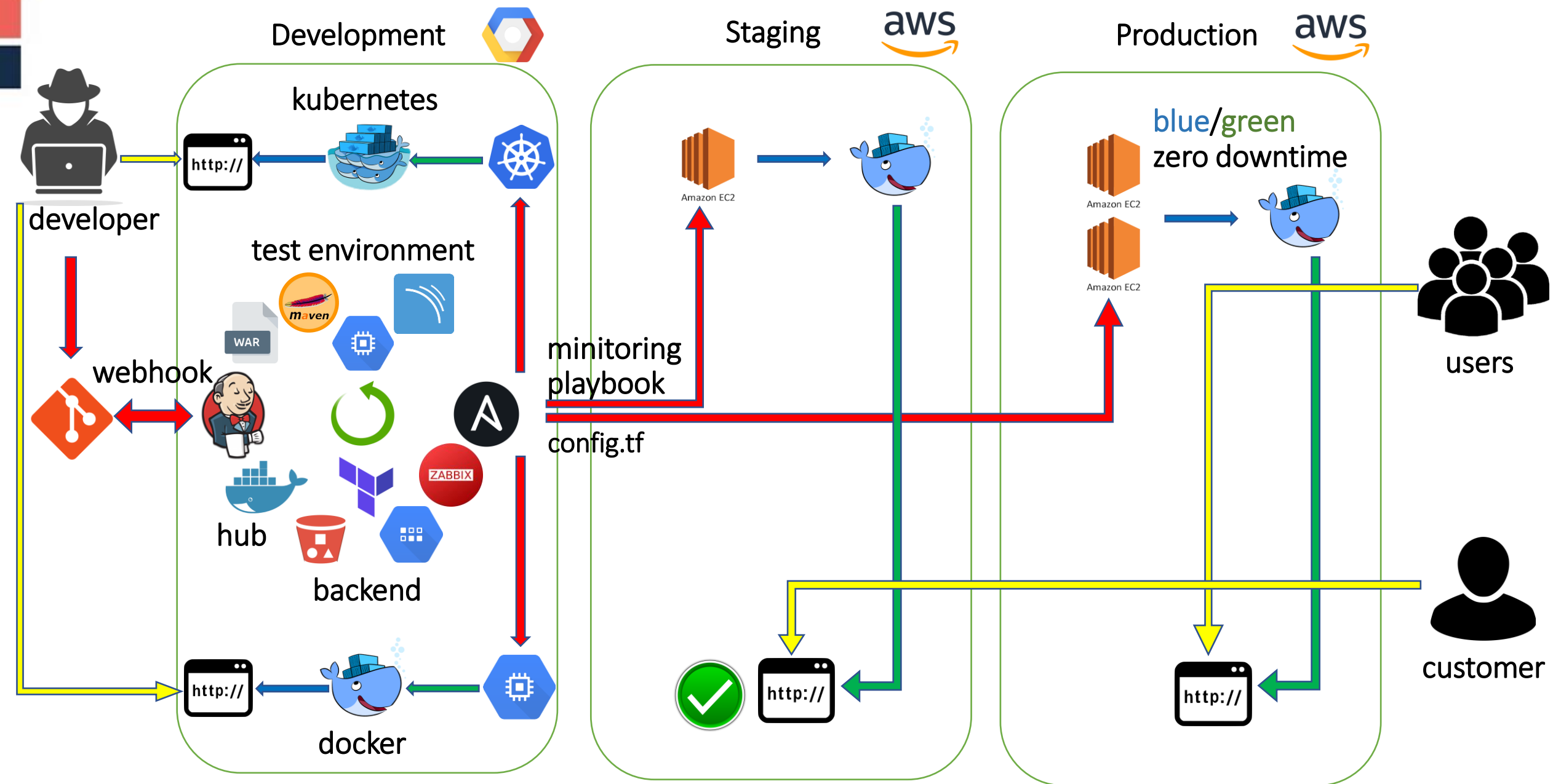
CentOS



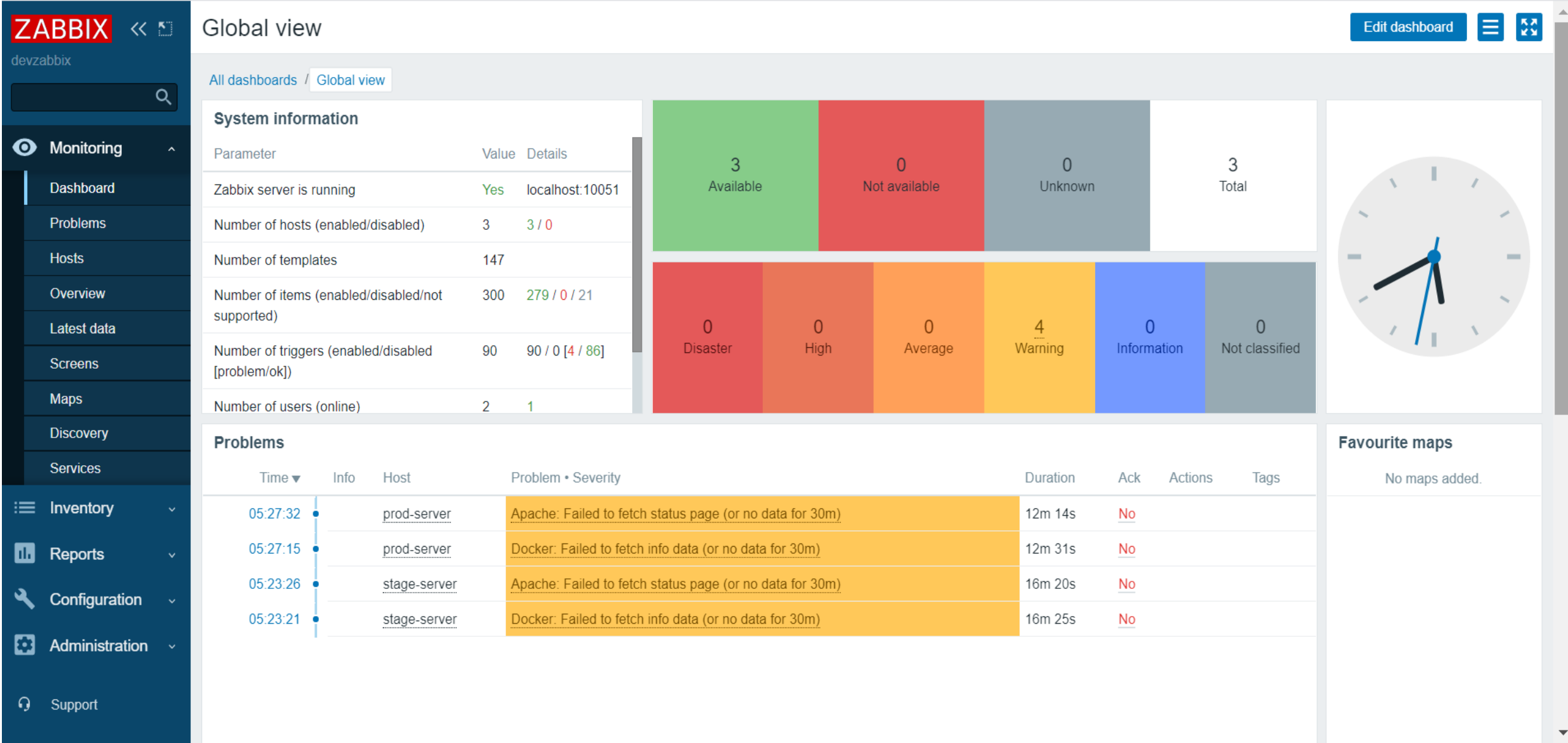
VirtualBox




# CI / CD process



# Zabbix monitoring



# SonarQube test




Projects


Issues

Rules

Quality Profiles

Quality Gates

 Search for projects, sub-projects and files...

 Log in

Filters

Search by project name or key

Quality Gate

Passed

3

Warning

0

Failed

0

Reliability

sort list byworstbest

A

3

B and worse

0

C and worse

0

D and worse

0

E

0

Security

sort list byworstbest

A

0

B and worse

3

C and worse

2

D and worse

0

List

Visualizations

3 projects

Maven Project

Quality Gate: 

Passed

A

Reliability

D

Security

A

Maintainability

0.0%

Coverage

0.0%

Duplications

XS

272

XML, Java

Last analysis on November 11, 2020 1:55 AM

TomcatDemoApp

Quality Gate: 

Passed

A

Reliability

B

Security

A

Maintainability

0.0%

Coverage

0.0%

Duplications

XS

116

Java, XML

Last analysis on November 8, 2020 7:41 PM

WebApp-1.0.0-SNAPSHOT

Quality Gate: 

Passed

A

Reliability

D

Security

A

Maintainability

0.0%

Coverage

0.0%

Duplications

XS

110

XML, Java

Last analysis on November 8, 2020 7:22 PM

3 of 3 shown

# Step 1. Deploy code or changes

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

```

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 .

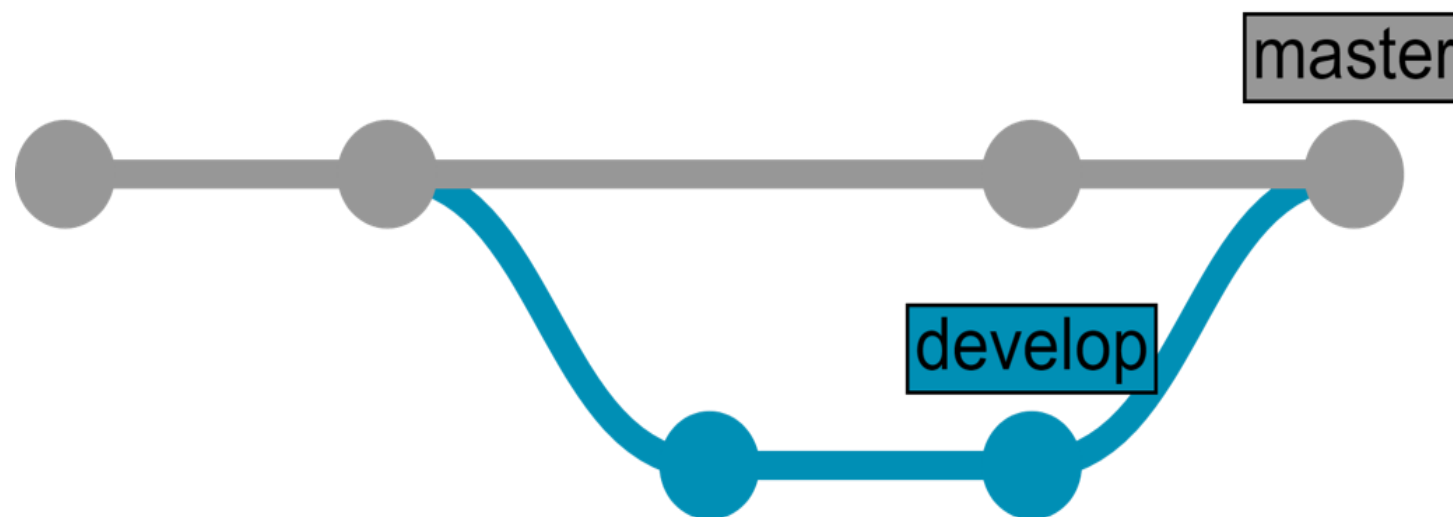
User@DESKTOP-6FBBSGI MINGW64 /c/git_epam/hello-world (master)
$ git commit -m "web v10"
On branch master
Your branch is ahead of 'origin/master' by 2 commits.
(use "git push" to publish your local commits)

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)
$

```



Webhook to Jenkins →

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

The screenshot shows the Jenkins web interface. On the left, there's a sidebar with navigation links: People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, My Views, Lockable Resources, and New View. The main area is divided into two sections: 'Build Queue' and 'Build Executor Status'.

**Build Queue:** This section shows a list of builds waiting to be executed. The builds are listed in a table with columns 'S' (Status), 'W' (Waiting), and 'Name'. The builds are:

S	W	Name
●	☀	DEV-CD-K8s-deploy-Image-to-cluster-using-Ansible-playbook
●	☀	DEV-CI-K8s-create-Image-using-Ansible-playbook
●	☀	DEV-K8s-create-GCP-Cluster-using-Ansible-playbook
●	☀	DEV-K8s-delete-GCP-Cluster-using-Ansible-playbook
●	☀	PROD-deploy-Container-on-AWS-using-Ansible-playbook
●	☀	STAGE-deploy-Container-on-AWS-using-Ansible-playbooks
●	☁	Test-deploy-on-Docker-Container-using-Ansible-playbook

The build 'Test-deploy-on-Docker-Container-using-Ansible-playbook' is highlighted with a red box.

**Build Executor Status:** This section shows the status of the build executors. There are three executors: 'master', 'slave-docker', and 'slave-kubernetes'. The 'slave-docker' executor is highlighted with a red box and shows two idle instances.

**Build Queue:** No builds in the queue.

**Build Executor Status:**

- master: 1 Idle
- slave-docker: 1 Idle, 2 Idle
- slave-kubernetes: 1 Idle, 2 Idle

## 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/>

Made by Volodymyr Verholyak

Make changes by git tester user

Version 1.3

# Hello World!

You are on the TomCat Server

[Servlet Examples with Code](#)

Deploy to Kubernetes →



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

People

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Lockable Resources

New View

**Build Queue**

No builds in the queue.

**Build Executor Status**

master

1 Idle

2 Idle

slave-docker

1 Idle

2 Idle

slave-kubernetes

1 Idle

2 Idle

S	W	Name ↓
●	☀	DEV-CD-K8s-deploy-Image-to-cluster-using-Ansible-playbook
●	☀	DEV-CI-K8s-create-Image-using-Ansible-playbook
●	☀	DEV-K8s-create-GCP-Cluster-using-Ansible-playbook
●	☀	DEV-K8s-delete-GCP-Cluster-using-Ansible-playbook
●	☀	PROD-deploy-Container-on-AWS-using-Ansible-playbook
●	☀	STAGE-deploy-Container-on-AWS-using-Ansible-playbooks
●	☁	Test-deploy-on-Docker-Container-using-Ansible-playbook

Icon: S M L

Apps ★ Bookmarks E Маршрутки Львов... » | Other bookmarks

Made by Volodymyr Verholyak

Make changes by git tester user

Version 1.3

## Hello World!

You are on the TomCat Server

[Servlet Examples with Code](#)

## 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)

The screenshot shows the Jenkins web interface. On the left, there's a sidebar with navigation links: People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, My Views, Lockable Resources, and New View. Below these are sections for 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status'. The 'Build Executor Status' section shows a list of executors: 'master' (1 Idle) and 'slave-docker' (2 Idle, highlighted with a red box). Below 'slave-docker' is 'slave-kubernetes' (1 Idle, 2 Idle). The main area displays a table of Jenkins jobs. The job 'STAGE-deploy-Container-on-AWS-using-Ansible-playbooks' is highlighted with a red box. Below the table, there's a preview of the application running on the slave-docker executor, showing a 'Hello World!' message and a link to 'Servlet Examples with Code'.

S	W	Name ↓
●	☀	DEV-CD-K8s-deploy-Image-to-cluster-using-Ansible-playbook
●	☀	DEV-CI-K8s-create-Image-using-Ansible-playbook
●	☀	DEV-K8s-create-GCP-Cluster-using-Ansible-playbook
●	☀	DEV-K8s-delete-GCP-Cluster-using-Ansible-playbook
●	☀	PROD-deploy-Container-on-AWS-using-Ansible-playbook
●	☀	<b>STAGE-deploy-Container-on-AWS-using-Ansible-playbooks</b>
●	☁	Test-deploy-on-Docker-Container-using-Ansible-playbook

Icon: S M L

Apps ★ Bookmarks E Маршрутки Львов... » Other bookmarks

Made by Volodymyr Verholyak  
Make changes by git tester user  
Version 1.3

## Hello World!

You are on the TomCat Server

[Servlet Examples with Code](#)

## 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)

The screenshot shows the Jenkins dashboard. On the left, there's a sidebar with navigation links: People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, My Views, Lockable Resources, and New View. The main area displays a table of build jobs. The job 'PROD-deploy-Container-on-AWS-using-Ansible-playbook' is highlighted with a red box. Below the table, there's a section for 'Build Queue' and 'Build Executor Status'. The 'Build Executor Status' section shows a list of executors, with 'slave-docker' highlighted by a red box, indicating it has 2 idle executors.

S	W	Name ↓
●	☀	DEV-CD-K8s-deploy-Image-to-cluster-using-Ansible-playbook
●	☀	DEV-CI-K8s-create-Image-using-Ansible-playbook
●	☀	DEV-K8s-create-GCP-Cluster-using-Ansible-playbook
●	☀	DEV-K8s-delete-GCP-Cluster-using-Ansible-playbook
●	☀	PROD-deploy-Container-on-AWS-using-Ansible-playbook
●	☀	STAGE-deploy-Container-on-AWS-using-Ansible-playbooks
●	☁	Test-deploy-on-Docker-Container-using-Ansible-playbook

Icon: S M L

Build Queue: No builds in the queue.

Build Executor Status:

- master: 1 Idle
- slave-docker: 2 Idle
- slave-kubernetes: 2 Idle

## 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/>

The screenshot shows a web application running on a TomCat Server. The page has a light blue background and contains the following text:


Made by Volodymyr Verholyak  
Make changes by git tester user  
Version 1.3

# Hello World!


You are on the TomCat Server

[Servlet Examples with Code](#)

Deploy to DEV using Jenkins →



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



# Infrastructure in pictures

A Kubernetes cluster is a managed group of VM instances for running containerized applications. [Learn more](#)

Filter by label or name						
<input type="checkbox"/> Name ^	Location	Cluster size	Total cores	Total memory	Notifications	Labels
<input type="checkbox"/> <input checked="" type="checkbox"/> demo-k8s	europa-west4-a	4	8 vCPUs	16.00 GB		<a href="#">Connect</a>



Instances (2) <a href="#">Info</a>						<a href="#">Refresh</a> <a href="#">Connect</a>
Filter instances						
<input type="checkbox"/>	Name ^	Instance ID	Instance state ^	Instance type ^	Status check	
<input type="checkbox"/>	PROD WebSe...	i-0f7f5b3b0412cc2c3	<input checked="" type="checkbox"/> Running	t2.micro	<input checked="" type="checkbox"/> 2/2 checks ...	
<input type="checkbox"/>	STAG WebSe...	i-07776e49c9cfd66d6	<input checked="" type="checkbox"/> Running	t2.micro	<input checked="" type="checkbox"/> 2/2 checks ...	

Elastic IP addresses (2)			
Filter Elastic IP addresses			
<input type="checkbox"/>	Name ^	Allocated IPv4 add... ^	Type ^
<input type="checkbox"/>	PROD WebServer IP	18.159.26.114	Public IP
<input type="checkbox"/>	STAGE WebServer IP	18.198.63.214	Public IP



Google Cloud Platform

<input type="checkbox"/>	Name	Zone	Internal IP	External IP	Connect
<input type="checkbox"/>	<input checked="" type="checkbox"/> gke-demo-k8s-default-pool-7201793d-0bvm	europa-west4-a	10.164.15.205 (nic0)	34.90.0.89	SSH
<input type="checkbox"/>	<input checked="" type="checkbox"/> gke-demo-k8s-default-pool-7201793d-wx6j	europa-west4-a	10.164.15.204 (nic0)	34.90.228.124	SSH
<input type="checkbox"/>	<input checked="" type="checkbox"/> gke-demo-k8s-default-pool-7201793d-xhx5	europa-west4-a	10.164.15.203 (nic0)	35.204.129.127	SSH
<input type="checkbox"/>	<input checked="" type="checkbox"/> gke-demo-k8s-default-pool-7201793d-zwg3	europa-west4-a	10.164.15.202 (nic0)	35.204.36.248	SSH
<input type="checkbox"/>	<input checked="" type="checkbox"/> server-docker-aws	europa-west4-a	10.164.0.10 (nic0)	35.214.209.115	SSH
<input type="checkbox"/>	<input checked="" type="checkbox"/> server-sonarqube	europa-west4-a	10.164.0.61 (nic0)	34.91.174.170	SSH

## Security Groups (3) [Info](#)

Filter security groups				
<input type="checkbox"/>	Name ^	Security group ID ^	Security group name ^	VPC ID
<input type="checkbox"/>	Dynamic Security G...	sg-04c0c85457d5aef00	PROD Dynamic SG	vpc-444f8d2e
<input type="checkbox"/>	Web Server Securit...	sg-0d182f37301f0d1d4	WebServer SG Stag	vpc-444f8d2e
<input type="checkbox"/>	my-own-sg	sg-1889667f	default	vpc-444f8d2e

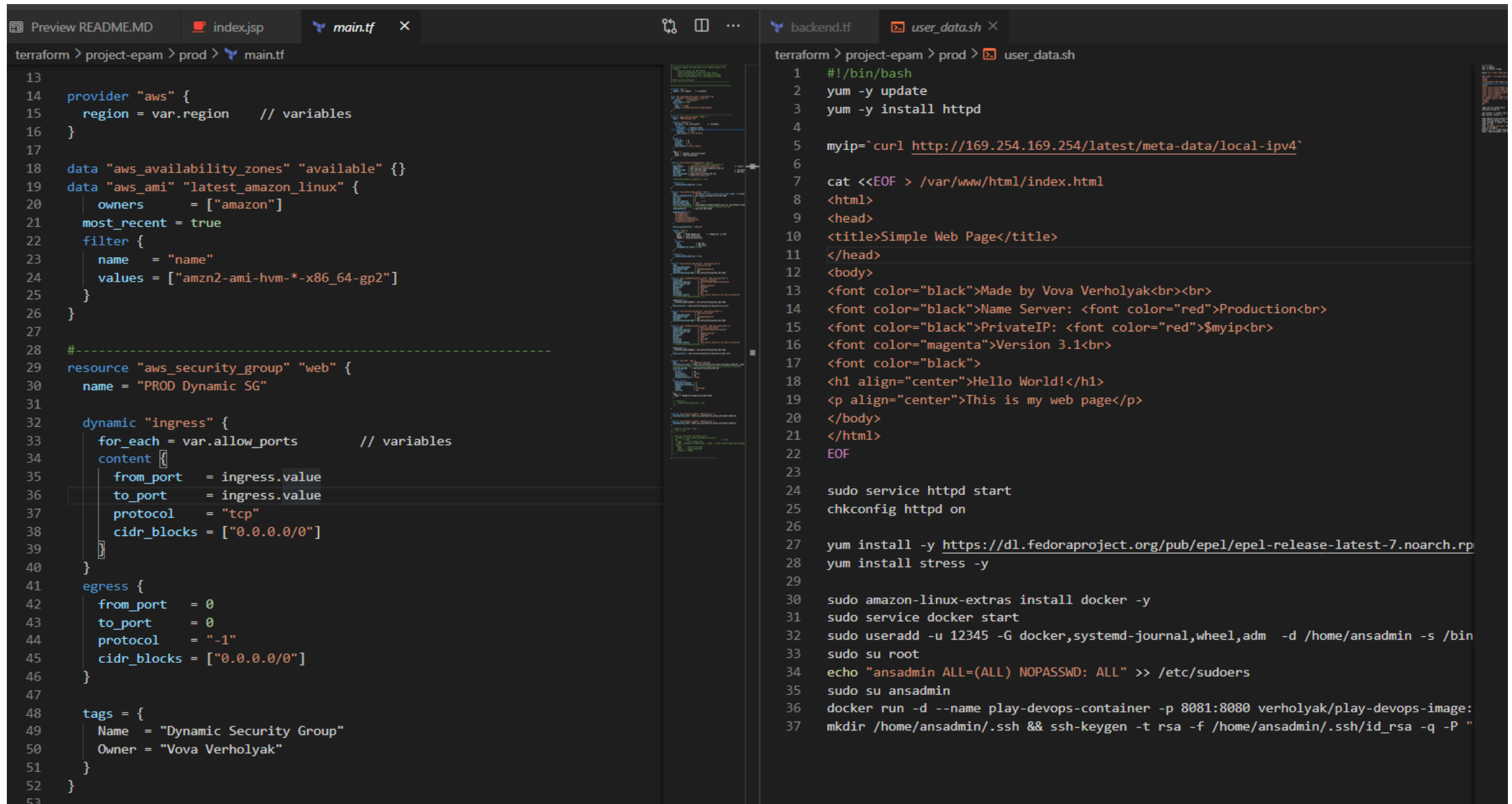
[Create Load Balancer](#) [Actions](#)

Filter by tags and attributes or search by keyword			
<input checked="" type="checkbox"/>	Name ^	DNS name ^	State ^
<input checked="" type="checkbox"/>	WebServer-HA-ELB	WebServer-HA-ELB-646341...	vpc-444f8d2e

## Auto Scaling groups (1)

Search your Auto Scaling groups				
<input type="checkbox"/>	Name ^	Launch templ... ^	Instances ^	Status ^
<input type="checkbox"/>	ASG-WebServer-t	WebServer-Highl...	1	-

# Terraform: infrastructure as code



The screenshot shows a code editor with two tabs: `main.tf` and `user_data.sh`. The `main.tf` tab is active, showing Terraform configuration for an AWS security group. The `user_data.sh` tab is also visible, showing a shell script for setting up a web server.

```
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
```

```
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.rpm
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](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 verholiyak/play-devops-image

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

    - name: remove docker image from ansible server
      command: docker rmi play-devops-image:latest verholiyak/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 verholiyak/play-devops-image:latest
      ignore_errors: yes

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

    - name: create container using play-devops-image
      command: docker run -d --name play-devops-container -p 8081:8080 verholiyak/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 ?

