

CILIST **(CILSY LIST)**

**Migrasi Infrastruktur On Premises
ke Cloud AWS**

Joko Sarjono Slameto

Problem

Problem 1

Develop new application based on MERN Stack (MongoDB, ExpressJS, ReactJS, dan NodeJS) on single VM instance

Problem 2

Have not implemented DevOps and CI/CD

Goals

1. Topology Design
2. Local Development.
3. Environment Staging(testing) dan Production(user)
4. Working flow and CI/CD
5. Centralized monitoring (kube-metric, Prometheus, Grafana) dan logging (Elasticsearch, Fluentd, Kibana)
6. Centralized source code repository dan scalable
7. Budgeting(6 bulan)
8. High Available



Pros and Cons using K8s



1

Pros:

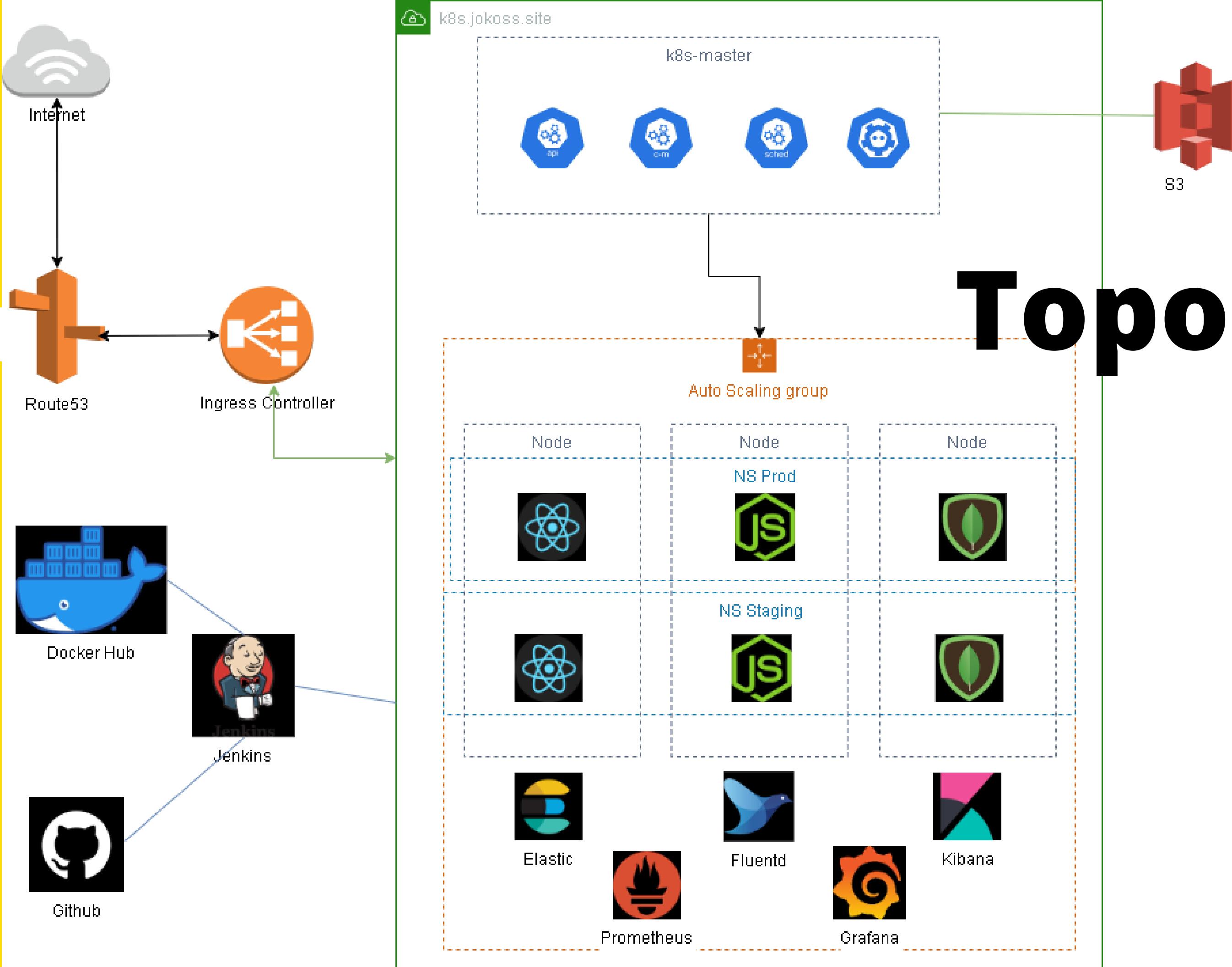
- Reusable yaml script
- Stable and scalable
- Docker provides lightweight and repeatable environments

2

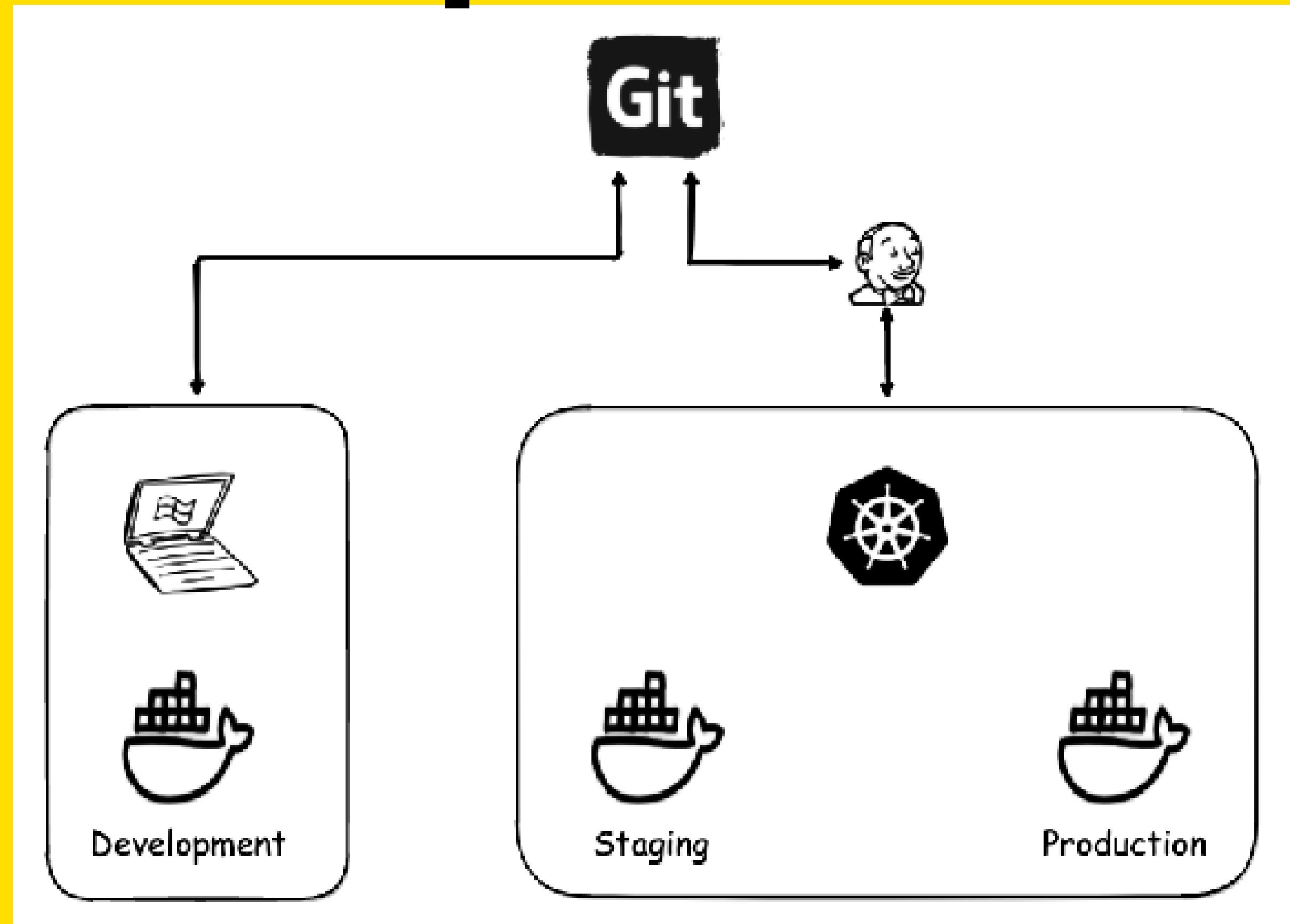
Cons :

- Might be overkill for small scale and simple apps
- Managing K8s is not really easy, so need more knowledges and experiences

Topology



Local Development



Budgeting (Minimum Load)

Service	Monthly (in USD)	6 months total (in USD)	Configuration summary
Amazon EC2	184.93	1109.58	Operating system (Linux)
			Quantity (4) - 3 worker node k8s and 1 jenkins master
			Pricing strategy (On-Demand Instances)
			Storage amount (30 GB)
			Instance type (t2.medium)
			Operating system (Linux)
Amazon EC2	88.86	533.16	Quantity (1) - 1 master node k8s
			Pricing strategy (On-Demand Instances)
			Storage amount (30 GB)
			Instance type (t2.large)
			Operating system (Linux)
			Quantity (1) - 1 bastion host
Amazon EC2	14.26	85.56	Pricing strategy (On-Demand Instances)
			Storage amount (30 GB)
			Instance type (t2.micro)
			Storage amount (50 GB), Type gp2
			S3 Standard storage (1 GB per month)
			Number of Classic Load Balancers (3), Processed bytes per CLB (10 GB per month)
Amazon Route 53	2.2	13.2	Hosted Zones (1)
AWS Data Transfer	1.08	6.48	DT Outbound: Internet (10 GB per month), Data transfer cost (1.08)
Total cost	363.67	2182.02	

Budgeting (Maximum Load)

Service	Monthly (in USD)	6 months total (in USD)	Configuration summary
Amazon EC2	277.39	1664.34	Operating system (Linux)
			Quantity (6) - 5 worker node k8s and 1 jenkins master
			Pricing strategy (On-Demand Instances)
			Storage amount (30 GB)
			Instance type (t2.medium)
			Operating system (Linux)
Amazon EC2	88.86	533.16	Quantity (1) - 1 master node k8s
			Pricing strategy (On-Demand Instances)
			Storage amount (30 GB)
			Instance type (t2.large)
			Operating system (Linux)
			Quantity (1) - 1 bastion host
Amazon EC2	14.26	85.56	Pricing strategy (On-Demand Instances)
			Storage amount (30 GB)
			Instance type (t2.micro)
			Storage amount (50 GB), Type gp2
			S3 Standard storage (1 GB per month)
			Number of Classic Load Balancers (3), Processed bytes per CLB (10 GB per month)
Amazon Route 53	2.2	13.2	Hosted Zones (1)
AWS Data Transfer	1.08	6.48	DT Outbound: Internet (10 GB per month), Data transfer cost (1.08)
Total cost	456.13	2736.78	

Setup Cluster k8s

```
1  *** Install kops ***
2 curl -LO https://github.com/kubernetes/kops/releases/download/$(curl -s https://api.github.com/repos/kubernetes/kops/releases/latest |
3 grep tag_name | cut -d '"' -f 4)/kops-linux-amd64
4 chmod +x kops-linux-amd64
5 sudo mv kops-linux-amd64 /usr/local/bin/kops

6  *** Install kubectl ***
7 curl -LO https://storage.googleapis.com/kubernetes-release/release/` curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt`/bin/linux/amd64/kubectl
8 chmod +x ./kubectl
9 sudo mv ./kubectl /usr/local/bin/kubectl
10 kubectl version --client

11
12 *** Export variabel yang nanti akan dibutuhkan ***
13 export bucket_name=k8s-jokoss-site
14 export KOPS_CLUSTER_NAME=k8s.jokoss.site
15 export KOPS_STATE_STORE=s3://${bucket_name}
16

17 *** Create cluster ***
18 kops create cluster --zones=ap-southeast-1a \
19 --node-count=3 \
20 --master-count=1 \
21 --node-size=t2.micro \
22 --master-size=t2.medium \
23 --name=${KOPS_CLUSTER_NAME} \
24 --ssh-public-key=~/.ssh/id_rsa.pub

25
26 kops update cluster --name ${KOPS_CLUSTER_NAME} --yes --admin
27 kops validate cluster --wait 10m
28 kubectl get nodes --show-labels
29
```

Install ingress and cert manager

***** Install Ingress *****

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/nginx-0.27.1/deploy/static/mandatory.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/nginx-0.27.1/deploy/static/provider/aws/service-l4.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/nginx-0.27.1/deploy/static/provider/aws/patch-configmap-l4.yaml
```

```
git clone https://github.com/kubernetes/ingress-nginx
cd ingress-nginx/deploy/static/provider/aws/
kubectl apply -f deploy.yaml
```

***** Create namespace *****

```
kubectl create namespace staging
kubectl create namespace production
```

***** Install Certificate Authority *****

```
snap install helm --classic
ln -s /snap/bin/helm /usr/local/bin/helm
kubectl apply --validate=false -f https://github.com/jetstack/cert-manager/releases/download/v0.14.3/cert-manager.crds.yaml
kubectl create namespace cert-manager
helm repo add jetstack https://charts.jetstack.io
helm repo update
helm --version
helm install cert-manager --namespace cert-manager --version v0.14.3 jetstack/cert-manager
kubectl get pods --namespace cert-manager
```

***** Setting node autoscaler *****

```
kops edit instancegroups nodes-ap-southeast-1a
```

*** Change this value**

```
  maxSize: 5
```

```
  minSize: 1
```

```
kops edit cluster
```

*** Add this value on spec**

```
spec:
```

```
  additionalPolicies:
```

```
    node: |
```

```
[
```

```
{
```

```
  "Effect": "Allow",
```

```
  "Action": [
```

```
    "autoscaling:DescribeAutoScalingGroups",
```

```
    "autoscaling:DescribeAutoScalingInstances",
```

```
    "autoscaling:SetDesiredCapacity",
```

```
    "autoscaling:DescribeLaunchConfigurations",
```

```
    "autoscaling:DescribeTags",
```

```
    "autoscaling:TerminateInstanceInAutoScalingGroup"
```

```
  ],
```

```
  "Resource": ["*"]
```

```
}
```

```
]
```

```
wget https://raw.githubusercontent.com/kubernetes/autoscaler/master/cluster-autoscaler/cloudprovider/aws/examples/cluster-autoscaler-autodiscover.yaml
nano cluster-autoscaler-autodiscover.yaml
# Change this value
spec:
  serviceAccountName: cluster-autoscaler
  containers:
    - image: asia.gcr.io/k8s-artifacts-prod/autoscaling/cluster-autoscaler:v1.15.6
      command:
        - ./cluster-autoscaler
        - --v=4
        - --stderrthreshold=info
        - --cloud-provider=aws
        - --skip-nodes-with-local-storage=false
        - --expander=least-waste
        - --nodes=1:5:nodes-ap-southeast-1a.k8s.jokoss.site
        - --node-group-auto-discovery=asg:tag=k8s.io/cluster-autoscaler/enabled,k8s.io/cluster-autoscaler/k8s.jokoss.site
      env:
        - name: AWS_REGION
          value: ap-southeast-1
      volumeMounts:
        - name: ssl-certs
          mountPath: /etc/ssl/certs/ca-bundle.crt #/etc/ssl/certs/ca-bundle.crt for Amazon Linux Worker Nodes
            readOnly: true
      imagePullPolicy: "Always"
  volumes:
    - name: ssl-certs
      hostPath:
        path: "/etc/ssl/certs/ca-bundle.crt"
```

*** Install EFK ***

```
cd efk/
kubectl apply -f efk-ns.yaml
cd elastic/
kubectl apply -f p-elasticsearch-svc.yaml
kubectl apply -f p-elasticsearch-statefulset.yaml
kubectl rollout status sts/es-cluster --namespace=kube-logging
kubectl port-forward es-cluster-0 9200:9200 --namespace=kube-logging
cd ../kibana/
kubectl apply -f p-kibana-dpy.yaml
kubectl rollout status deployment/kibana --namespace=kube-logging
kubectl get pods --namespace=kube-logging
kubectl apply -f p-p-ekf-ingress.yaml
kubectl apply -f p-fluentd-dpy.yaml
kubectl get ds --namespace=kube-logging
kubectl get svc --namespace=kube-logging
kubectl get ing --namespace=kube-logging
kubectl apply -f p-counter-pod.yaml
```

← → C https://efk.jokoss.site/app/kibana#/discover?_g=0&_a=(columns:_source,index:af5282d0-97e4-11eb-ac02-b971447ddb72,interval:auto,query:(language:kuery,query:""),sort:[])&_t=1618000000000

D Discover

Help us improve the Elastic Stack by providing usage statistics for basic features. We will not share this data outside of Elastic. Read more

Yes No

12,782 hits

New Save Open Share Inspect

Filters Search KQL Last 15 minutes Show dates Refresh

+ Add filter

logstash-*

Apr 11, 2021 @ 18:17:51.680 - Apr 11, 2021 @ 18:32:51.680 — Auto

Selected fields

? _source

Available fields

@timestamp _id _index # _score _type

Count

1,000

500

0

18:18:00 18:19:00 18:20:00 18:21:00 18:22:00 18:23:00 18:24:00 18:25:00 18:26:00 18:27:00 18:28:00 18:29:00 18:30:00 18:31:00 18:32:00

@timestamp per 30 seconds

Time _source

Apr 11, 2021 @ 18:32:49.760 log: I0411 11:32:49.760088 1 static_autoscaler.go:187] Starting main loop stream: stderr

~~*** Install Prometheus Grafana ***~~

```
kubectl apply -f p-monitoring-pvc.yml  
kubectl apply -f p-monitoring-ns.yml  
kubectl apply -f p-prometheus-cr.yml  
kubectl apply -f p-prometheus-configmap.yml  
kubectl apply -f p-prometheus-deployment.yaml  
kubectl apply -f p-prometheus-service.yaml  
kubectl apply -f p-metrics-sa.yml  
kubectl apply -f p-metrics-rbac.yml  
kubectl apply -f p-metrics-cr.yml  
kubectl apply -f p-metrics-dpy.yml  
kubectl apply -f p-metrics-svc.yml  
kubectl apply -f p-grafana-datasource-config.yaml  
kubectl apply -f p-grafana-dpy.yaml  
kubectl apply -f p-grafana-svc.yml  
kubectl apply -f p-grafana-ingress.yml  
kubectl apply -f p-grafana-ingress.yml  
kubectl get ing -n monitoring
```



***** Install Jenkins Master *****

```
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
sudo apt-get update
sudo apt-get install openjdk-8-jdk
/var/lib/jenkins/secrets/initialAdminPassword
sudo apt install nginx software-properties-common
sudo openssl dhparam -out /etc/nginx/ssl/dhparam.pem 2048
sudo service nginx stop
sudo rm -rf /etc/nginx/sites-enabled/default
sudo add-apt-repository ppa:certbot/certbot
sudo apt install certbot
sudo certbot certonly --standalone -d jenkins.jokoss.site
systemctl restart nginx
sudo openssl dhparam -out /etc/nginx/ssl/dhparam.pem 2048
sudo vim /etc/nginx/sites-available/jenkins.jokoss.site
sudo ln -s /etc/nginx/sites-available/jenkins.jokoss.site /etc/nginx/sites-enabled/jenkins.jokoss.site
sudo nginx -t
sudo systemctl restart nginx
snap install docker
docker login
curl -LO https://storage.googleapis.com/kubernetes-release/release/curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt/bin/linux/amd64/kubectl
chmod +x ./kubectl
sudo mv ./kubectl /usr/local/bin/kubectl
```

← → ⌂ 🔒 jenkins.jokoss.site/job/mern-todo-prod/configure

Dashboard > mern-todo-prod >

General Build Triggers Advanced Project Options Pipeline

Discard old builds

Do not allow concurrent builds

Do not allow the pipeline to resume if the master restarts

GitHub project

Project url <https://github.com/jokoss92/mern-todo-app/> Advanced...

Pipeline speed/durability override

Preserve stashes from completed builds

This project is parameterized

Throttle builds

Build Triggers

Build after other projects are built

Build periodically

GitHub hook trigger for GITScm polling

Poll SCM

Disable this project

Quiet period

Save **Apply** (m scripts)

Dashboard ➔ mern-todo-prod ➔

General Build Triggers

Advanced Project Options

Pipeline

Definition

Pipeline script from SCM

SCM

Git

Repositories

Repository URL

https://github.com/jokoss92/mern-todo-app.git

Credentials

jokoss92/*********

Add

?

Advanced...

Add Repository

Branches to build

Save

Apply

(any)

X

?



Search or jump to...



Pull requests Issues Marketplace Explore

jokoss92 / **mern-todo-app**

forked from sdcilsy/mern-todo-app

Watch ▾ 0

Star 0

Fork 9



Code



Pull requests



Actions



Projects



Wiki



Security



Insights



Settings

Options

Manage access

Security & analysis

Branches

Webhooks

Notifications

Integrations

Deploy keys

Actions

Environments

Secrets

Webhooks

[Add webhook](#)

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

<https://jenkins.jokoss.site/github-webhook-test/> (push)[Edit](#)[Delete](#)

```
backend.yml frontend.yml step-by-step.md Jenkinsfile X

mern-todo-app > Jenkinsfile
 1  env.DOCKER_REGISTRY = 'johnojss'
 2  env.DOCKER_IMAGE_FRONTEND = 'mern-frontend'
 3  env.DOCKER_IMAGE_BACKEND = 'mern-backend'
 4
 5  pipeline {
 6      agent any
 7      stages {
 8          stage('Hello World') {
 9              steps {
10                  sh "whoami"
11              }
12          }
13          stage('Git clone from Github') {
14              steps {
15                  sh "git clone https://github.com/jokoss92/mern-todo-app.git"
16              }
17          }
18          stage('Docker Build Image') {
19              steps {
20                  sh "cd mern-todo-app && docker build -t $DOCKER_REGISTRY/$DOCKER_IMAGE_FRONTEND:${BUILD_NUMBER} ."
21                  sh "cd mern-todo-app/backend && docker build -t $DOCKER_REGISTRY/$DOCKER_IMAGE_BACKEND:${BUILD_NUMBER} ."
22              }
23          }
24          stage('Push Docker Image') {
25              steps {
26                  sh "docker push $DOCKER_REGISTRY/$DOCKER_IMAGE_FRONTEND:${BUILD_NUMBER}"
27                  sh "docker push $DOCKER_REGISTRY/$DOCKER_IMAGE_BACKEND:${BUILD_NUMBER}"
28              }
29          }
30          stage('Deploy Image to Kubernetes') {
31              steps {
32                  sh'''cd mern-todo-app && sed -i "15d" frontend.yml'''
33                  sh'''cd mern-todo-app && sed -i "14 a \\\'\\\' image: $DOCKER_REGISTRY/$DOCKER_IMAGE_FRONTEND:${BUILD_NUMBER} \\\'\\\'" frontend.yml'''
34                  sh "cd mern-todo-app && kubectl apply -f frontend.yml"
35                  sh'''cd mern-todo-app && sed -i "15d" backend.yml'''
36                  sh'''cd mern-todo-app && sed -i "14 a \\\'\\\' image: $DOCKER_REGISTRY/$DOCKER_IMAGE_BACKEND:${BUILD_NUMBER} \\\'\\\'" backend.yml'''
37                  sh "cd mern-todo-app && kubectl apply -f backend.yml"
38              }
39          }
40          stage('Delete Image') {
```

```
3  
4  mongoose.connect('mongodb://db.jokoss.site:27017/todos', { useNewUrlParser: true });  
5  const connection = mongoose.connection;  
6
```

```
    axios.post('https://be.jokoss.site/todos/add', newTodo)  
      .then(res => console.log(res.data));
```

```
componentDidMount() {  
  axios.get('https://be.jokoss.site/todos/'+this.props.match.params.id)  
    .then(response => {  
      this.setState({  
        todo_description: response.data.todo_description,  
        todo_responsible: response.data.todo_responsible,  
        todo_priority: response.data.todo_priority,  
        todo_completed: response.data.todo_completed  
      })  
    })  
    .catch(function (error) {  
      console.log(error);  
    })  
}
```

```
componentDidMount() {  
  axios.get('https://be.jokoss.site/todos/')    .then(response => {  
      this.setState({ todos: response.data });  
    })  
    .catch(function (error){  
      console.log(error);  
    })  
}
```

```
mern-todo-app > ! mongoDB.yaml
 1  apiVersion: v1
 2  kind: PersistentVolumeClaim
 3  metadata:
 4    name: pvc-mongodb
 5    namespace: production
 6  spec:
 7    accessModes:
 8      - ReadWriteOnce
 9    resources:
10      requests:
11        storage: 20Gi
12    storageClassName: gp2
13
14  ---
15
16  apiVersion: apps/v1
17  kind: Deployment
18  metadata:
19    name: mongodb
20    namespace: production
21    labels:
22      app: mongodb
23  spec:
24    replicas: 1
25    template:
26      metadata:
27        labels:
28          app: mongodb
29    spec:
30      containers:
31        - name: mongodb
32          image: mongo:latest
33          ports:
34            - containerPort: 27017
35          volumeMounts:
36            - mountPath: '/data/db'
37              name: mongo-storage
38        volumes:
39          - name: mongo-storage
40          persistentVolumeClaim:
41            claimName: pvc-mongodb
42    selector:
43      matchLabels:
44        app: mongodb
45  ---
46
47  apiVersion: v1
48  kind: Service
49  metadata:
50    name: mongodb-service
```

```
n-todo-app > ! backend.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: backend-deployment
  namespace: production
spec:
  replicas: 2
  template:
    metadata:
      labels:
        app: server
    spec:
      containers:
        - name: server
          image: johnojs/mern-backend
          ports:
            - containerPort: 4000
              name: nodejs-port
      selector:
        matchLabels:
          app: server
  ---
  apiVersion: v1
  kind: Service
  metadata:
    name: backend-service
    namespace: production
  spec:
    ports:
      - port: 4000
        protocol: TCP
    selector:
      app: server
    # type: LoadBalancer
  ---
  apiVersion: extensions/v1beta1
  kind: Ingress
  metadata:
    name: backend-ingress
    namespace: production
  annotations:
    nginx.ingress.kubernetes.io/rewrite-target: /
    kubernetes.io/ingress.class: "nginx"
    cert-manager.io/issuer: "backend-prod"
  spec:
    rules:
      - host: be.jokoss.site
        http:
```

```
mern-todo-app > ! frontend.yaml
 1  apiVersion: apps/v1
 2  kind: Deployment
 3  metadata:
 4    name: frontend-deployment
 5    namespace: production
 6  spec:
 7    replicas: 2
 8    template:
 9      metadata:
10        labels:
11          app: client
12      spec:
13        containers:
14          - name: client
15            image: johnojss/mern-frontend
16            ports:
17              - containerPort: 3000
18                name: http-port
19            selector:
20              matchLabels:
21                app: client
22
23  ---
24  apiVersion: v1
25  kind: Service
26  metadata:
27    name: frontend-service
28    namespace: production
29  spec:
30    ports:
31      - port: 80
32        protocol: TCP
33        targetPort: http-port
34    selector:
35      app: client
36    # type: LoadBalancer
37
38  ---
39  apiVersion: cert-manager.io/v1alpha2
40  kind: Issuer
41  metadata:
42    name: frontend-prod
43    namespace: production
44  spec:
45    acme:
46      # The ACME server URL
47      server: https://acme-v02.api.letsencrypt.org/directory
48      # Email address used for ACME registration
49      email: johno.smakaduta@gmail.com
50
```

<input type="checkbox"/>	Record name	Type	Routing	Differ...	Value/Route traffic to	
<input type="checkbox"/>	jokoss.site	A	Simple	-	54.251.28.56	
<input type="checkbox"/>	jokoss.site	NS	Simple	-	ns-964.awsdns-56.net. ns-1996.awsdns-57.co.uk. ns-92.awsdns-11.com. ns-1493.awsdns-58.org.	
<input type="checkbox"/>	jokoss.site	SOA	Simple	-	ns-964.awsdns-56.net. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400	
<input type="checkbox"/>	be.jokoss.site	CNAME	Simple	-	a465ff232e13949faa9b6b478cbb4dd5-518715906.ap-southeast-1.elb.amazonaws.com	
<input type="checkbox"/>	db.jokoss.site	CNAME	Simple	-	a92a1062ac1bf4fbb908d7f8fcadd8cc-399079979.ap-southeast-1.elb.amazonaws.com	
<input type="checkbox"/>	efk.jokoss.site	CNAME	Simple	-	a465ff232e13949faa9b6b478cbb4dd5-518715906.ap-southeast-1.elb.amazonaws.com	
<input type="checkbox"/>	fe.jokoss.site	CNAME	Simple	-	a465ff232e13949faa9b6b478cbb4dd5-518715906.ap-southeast-1.elb.amazonaws.com	
<input type="checkbox"/>	graf.jokoss.site	CNAME	Simple	-	a465ff232e13949faa9b6b478cbb4dd5-518715906.ap-southeast-1.elb.amazonaws.com	
<input type="checkbox"/>	jenkins.jokoss.site	A	Simple	-	52.77.224.61	
<input type="checkbox"/>	api.k8s.jokoss.site	A	Simple	-	13.212.24.214	
<input type="checkbox"/>	api.internal.k8s.jokoss.site	A	Simple	-	172.20.39.241	
<input type="checkbox"/>	kops-controller.internal.k8s.jokoss.site	A	Simple	-	172.20.39.241	
<input type="checkbox"/>	logs.jokoss.site	A	Simple	-	18.141.216.204	
<input type="checkbox"/>	mon.jokoss.site	CNAME	Simple	-	a465ff232e13949faa9b6b478cbb4dd5-518715906.ap-southeast-1.elb.amazonaws.com	

Pipeline mern-todo-prod

 Add description

Disable Project



Recent Changes

Stage View

Declarative: Checkout SCM	Hello World	Git clone from GitHub
Average stage times: (Average full run time: ~39s)	3s	312ms
#20 Apr 11 18:22 1 commit	2s	311ms



Console Output

```
Started by GitHub push by jokoss92
Obtained Jenkinsfile from git https://github.com/jokoss92/mern-todo-app.git
Running in Durability level: MAX_SURVIVABILITY
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/mern-todo-prod
[Pipeline] {
[Pipeline] stage
[Pipeline] {
  (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
using credential d1804f6f-e09b-4a70-953e-9ac24c119c7c
Cloning the remote Git repository
Cloning repository https://github.com/jokoss92/mern-todo-app.git
> git init /var/lib/jenkins/workspace/mern-todo-prod # timeout=10
Fetching upstream changes from https://github.com/jokoss92/mern-todo-app.git
> git --version # timeout=10
> git --version # 'git version 2.17.1'
using GIT_ASKPASS to set credentials
> git fetch --tags --progress -- https://github.com/jokoss92/mern-todo-app.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/jokoss92/mern-todo-app.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
```

fe.jokoss.site

Cilsy Todo App Todos Create Todo

Todos List

Description	Responsible	Priority	Action
Joko	Joko123	High	Edit
Levy	Ackerman	High	Edit

← → C 🔒 be.jokoss.site/todos

```
1 // 20210411182402
2 // https://be.jokoss.site/todos
3
4 [
5   {
6     "_id": "6072c9fa7045c100118adf2b",
7     "todo_description": "Joko",
8     "todo_responsible": "Joko123",
9     "todo_priority": "High",
10    "todo_completed": false,
11    "__v": 0
12  },
13  {
14    "_id": "6072dc49396006001880b285",
15    "todo_description": "Levy",
16    "todo_responsible": "Ackerman",
17    "todo_priority": "High",
18    "todo_completed": false,
19    "__v": 0
20  }
21 ]
```