

A decorative graphic on the left side of the slide. It consists of a blue parallelogram and a light green parallelogram, both tilted at an angle. The blue shape is in the foreground, and the green shape is partially behind it. They are set against a dark blue background with faint, lighter blue diagonal stripes.

DevOps Final Capstone Project

AGENDA

Fundamentals

Github & Development environment
Architectural Diagram

Infrastructure-as-Code

Terraform main.tf
Terraform Modules - Jenkins

Container Technologies

Dockerfile
Dockerhub Repo
Kubernetes
k9s

CI/CD

Demo
Github & Jenkins
Jenkinsfile
Webapp deployment

Observability systems

Fluentd, prometheus, grafana
Grafana dashboard and query

Fundamentals



Fundamentals



<https://github.com/Veronaz/DevopsFinalCapstonProject>

Development Env

see README.md for details

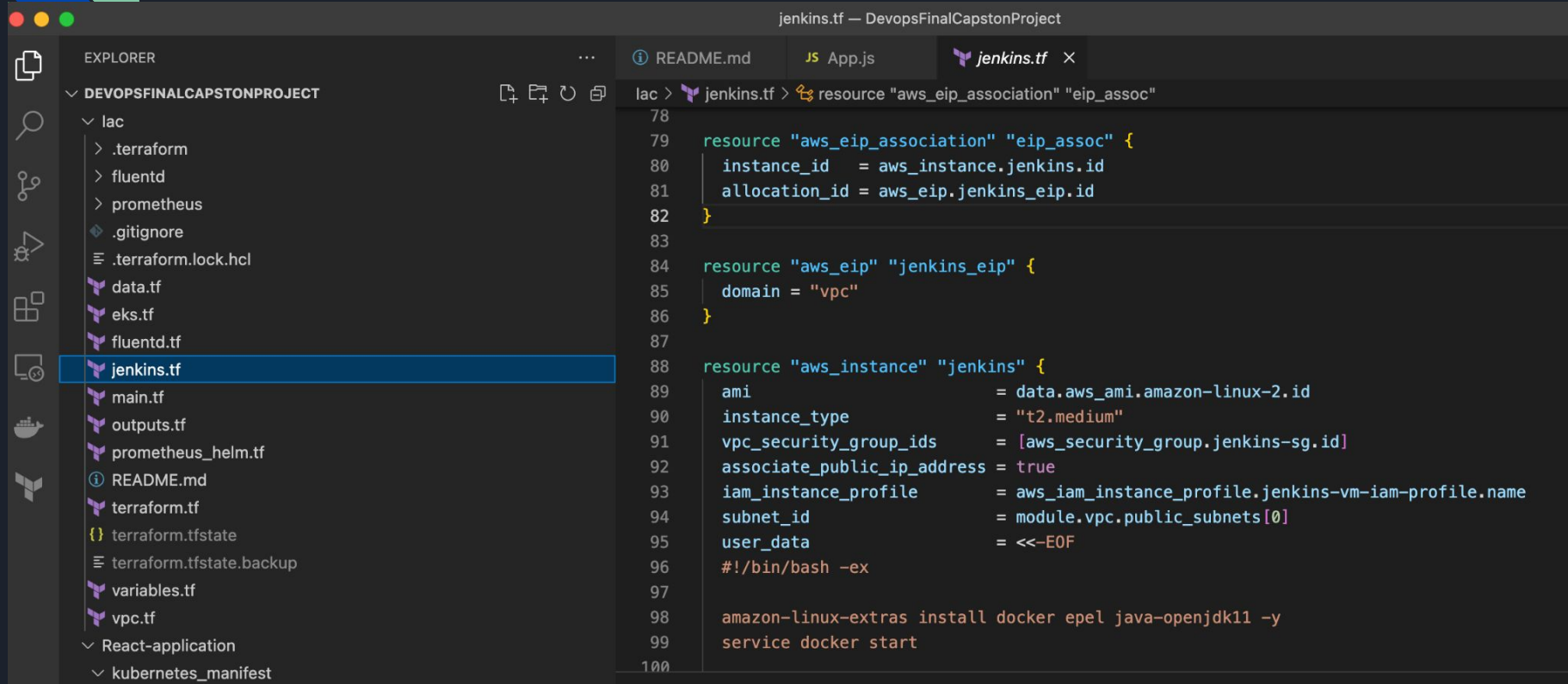
1. Awscli
2. Git
3. Terraform "tfswitch"
4. Node.js
5. Kubectl
6. Session manager
7. Docker

Infrastructure -as-Code

**Use infrastructure-as-Code technology
(e.g. Terraform) to deploy infrastructure**

1. Use infrastructure-as-code configuration
2. files for declarative deployment of infrastructure
3. Use configuration management tools such as Ansible to automate configuration of remote servers

Terraform Modules - Jenkins



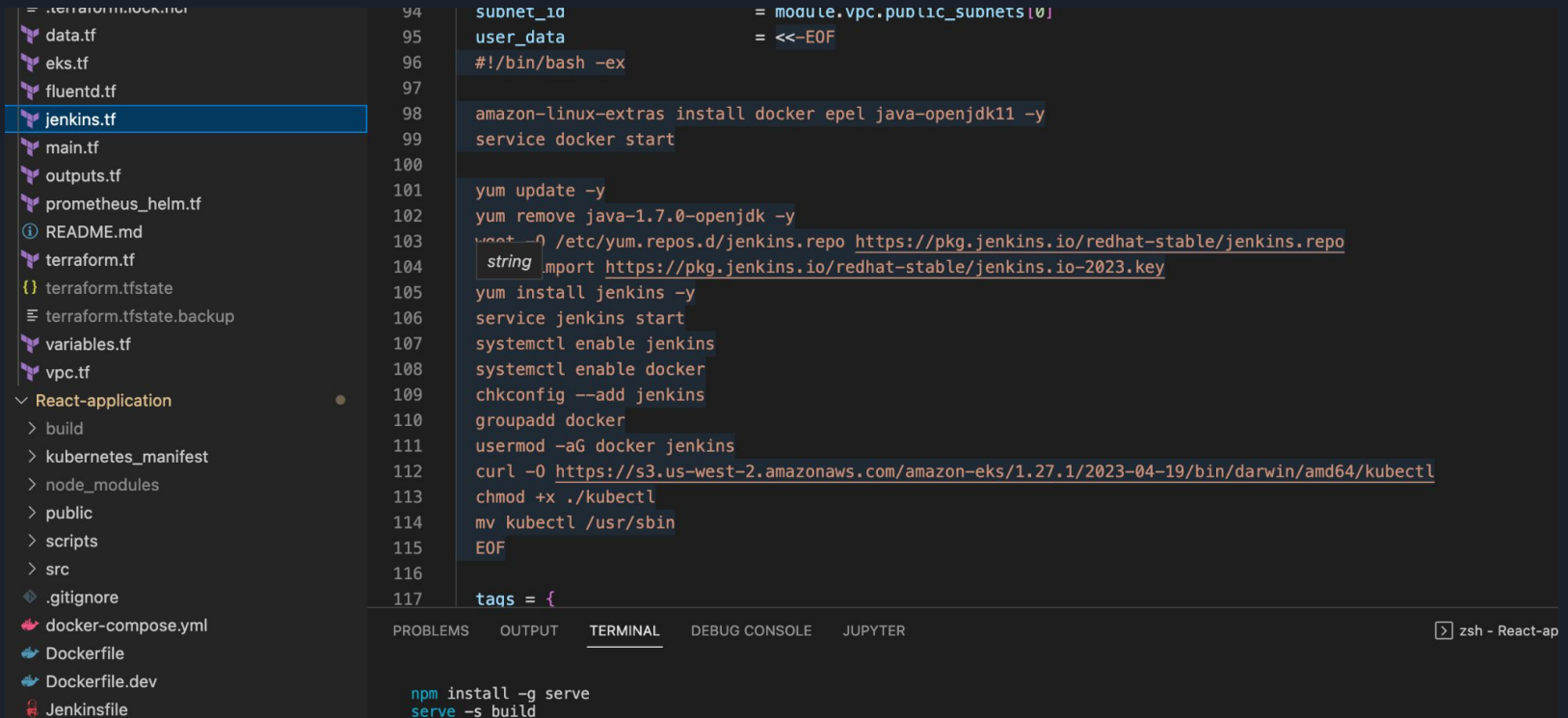
The screenshot displays a code editor window titled "jenkins.tf — DevopsFinalCapstonProject". The left sidebar shows the "EXPLORER" view with the following file structure:

- DEVOPSFINALCAPSTONPROJECT
 - lac
 - .terraform
 - fluentd
 - prometheus
 - .gitignore
 - .terraform.lock.hcl
 - data.tf
 - eks.tf
 - fluentd.tf
 - jenkins.tf**
 - main.tf
 - outputs.tf
 - prometheus_helm.tf
 - README.md
 - terraform.tf
 - terraform.tfstate
 - terraform.tfstate.backup
 - variables.tf
 - vpc.tf
 - React-application
 - kubernetes_manifest

The main editor shows the content of "jenkins.tf":

```
78 lac > jenkins.tf > resource "aws_eip_association" "eip_assoc"
79
80 resource "aws_eip_association" "eip_assoc" {
81     instance_id = aws_instance.jenkins.id
82     allocation_id = aws_eip.jenkins_eip.id
83 }
84
85 resource "aws_eip" "jenkins_eip" {
86     domain = "vpc"
87 }
88
89 resource "aws_instance" "jenkins" {
90     ami = data.aws_ami.amazon-linux-2.id
91     instance_type = "t2.medium"
92     vpc_security_group_ids = [aws_security_group.jenkins-sg.id]
93     associate_public_ip_address = true
94     iam_instance_profile = aws_iam_instance_profile.jenkins-vm-iam-profile.name
95     subnet_id = module.vpc.public_subnets[0]
96     user_data = <<-EOF
97     #!/bin/bash -ex
98     amazon-linux-extras install docker epel java-openjdk11 -y
99     service docker start
100 }
```

Terraform Modules - Jenkins



The image shows a VS Code editor interface with a file explorer on the left and a terminal window at the bottom.

File Explorer:

- terraform.tf
- data.tf
- eks.tf
- fluentd.tf
- jenkins.tf** (selected)
- main.tf
- outputs.tf
- prometheus_helm.tf
- README.md
- terraform.tf
- terraform.tfstate
- terraform.tfstate.backup
- variables.tf
- vpc.tf
- React-application
 - build
 - kubernetes_manifest
 - node_modules
 - public
 - scripts
 - src
 - .gitignore
 - docker-compose.yml
 - Dockerfile
 - Dockerfile.dev
 - Jenkinsfile

Terminal Window:

```
94 subnets = module.vpc.public_subnets[0]
95 user_data = <<-EOF
96 #!/bin/bash -ex
97
98 amazon-linux-extras install docker epel java-openjdk11 -y
99 service docker start
100
101 yum update -y
102 yum remove java-1.7.0-openjdk -y
103 wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo
104 string import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
105 yum install jenkins -y
106 service jenkins start
107 systemctl enable jenkins
108 systemctl enable docker
109 chkconfig --add jenkins
110 groupadd docker
111 usermod -aG docker jenkins
112 curl -O https://s3.us-west-2.amazonaws.com/amazon-eks/1.27.1/2023-04-19/bin/darwin/amd64/kubect
113 chmod +x ./kubectl
114 mv kubectl /usr/sbin
115 EOF
116
117 tags = {
```

Terminal Output:

```
npm install -g serve
serve -s build
```

Terminal Tabs: PROBLEMS, OUTPUT, TERMINAL, DEBUG CONSOLE, JUPYTER

Terminal Title Bar: zsh - React-ap

Container Technologies

Build optimized Docker images

1. Deploy apps in Kubernetes cluster
2. Leverage core Kubernetes resources
 - a. Deployments
 - b. Services
 - c. Load balancers
 - d. Volumes
 - e. Ingress
3. Leverage cloud managed Kubernetes services - e.g. AWS EKS



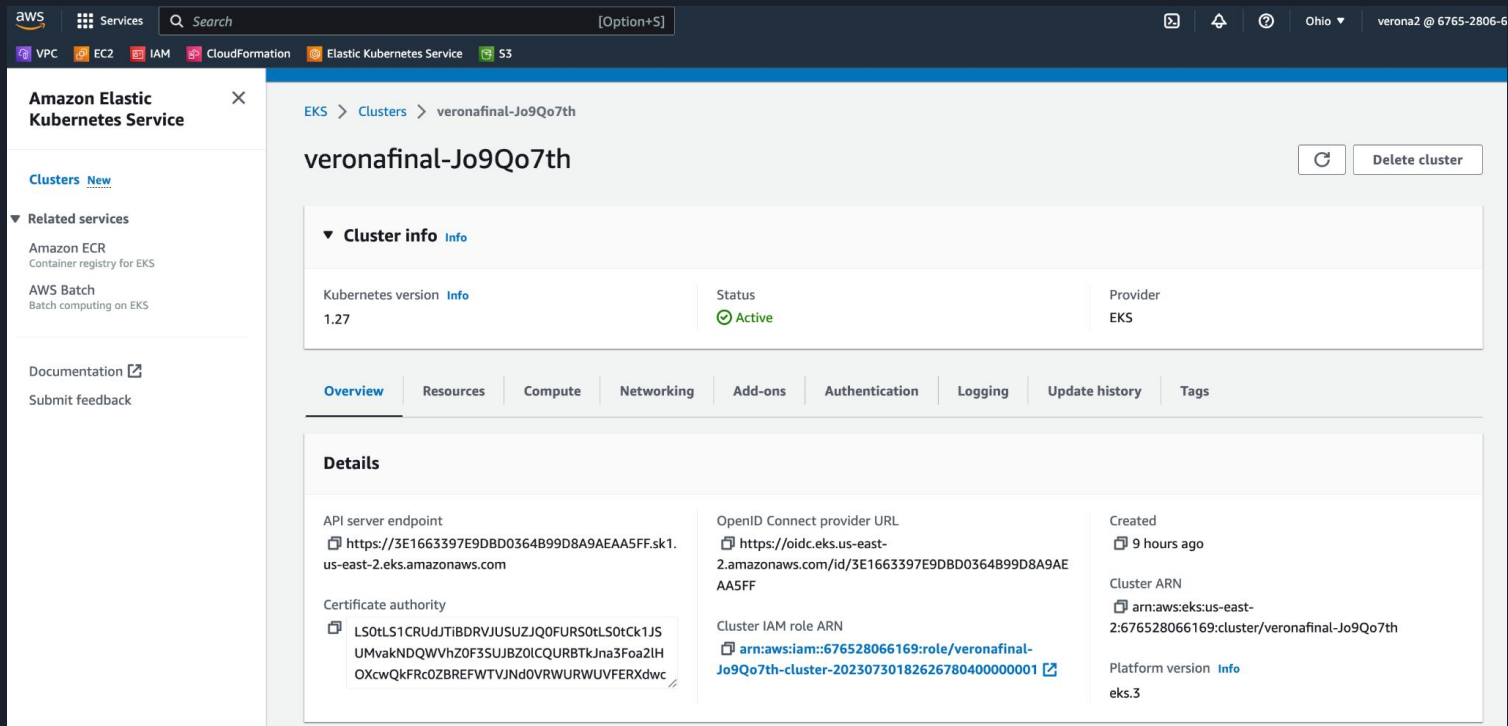
Dockerfile

Multiple stage build

React-application >  Dockerfile > ...

```
1 FROM node:16-alpine as base
2 WORKDIR '/app'
3 COPY package.json .
4 RUN npm install
5 COPY . .
6 RUN npm run build
7
8 FROM base as test
9 ENTRYPOINT npm run test
10
11 FROM nginx as runtime
12 COPY --from=base /app/build /usr/share/nginx/html
```

Kubernetes (AWS-EKS) Cluster



The screenshot displays the AWS Management Console interface for an Amazon Elastic Kubernetes Service (EKS) cluster. The top navigation bar includes the AWS logo, a 'Services' menu, a search bar, and the user's profile 'verona2 @ 6765-2806-6'. The left sidebar shows the 'Amazon Elastic Kubernetes Service' console, with options for 'Clusters', 'Related services' (Amazon ECR, AWS Batch), 'Documentation', and 'Submit feedback'.

The main content area shows the cluster 'veronafinal-Jo9Qo7th' with a 'Delete cluster' button. Below this, the 'Cluster info' section displays the following details:

Property	Value
Kubernetes version	1.27
Status	Active
Provider	EKS

The 'Details' section provides further information about the cluster:

Property	Value
API server endpoint	https://3E1663397E9DBD0364B99D8A9AEA5FF.sk1.us-east-2.eks.amazonaws.com
Certificate authority	LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSUMvakNDQWVhZ0F3SUJBZ0lCQURBTklna3Foa2lHOXcwQkFRc0ZBREFWTVJNd0VRWURWUVFERXdw
OpenID Connect provider URL	https://oidc.eks.us-east-2.amazonaws.com/id/3E1663397E9DBD0364B99D8A9AEA5FF
Cluster IAM role ARN	arn:aws:iam::676528066169:role/veronafinal-Jo9Qo7th-cluster-20230730182626780400000001
Created	9 hours ago
Cluster ARN	arn:aws:eks:us-east-2:676528066169:cluster/veronafinal-Jo9Qo7th
Platform version	eks.3

Kubernetes (AWS-EKS)

Nodes

Amazon Elastic
Kubernetes Service

Clusters [New](#)

Related services

Amazon ECR

Container registry for EKS

AWS Batch

Batch computing on EKS

Documentation

Submit feedback

► Cluster info [Info](#)

OverviewResourcesComputeNetworkingAdd-onsAuthenticationLoggingUpdate historyTags

Nodes (2) [Info](#)


Node name	Instance type	Node group	Created	Status
ip-10-0-2-139.us-east-2.compute.internal	t3.medium	node-group-1-20230730183801533800000013	Created 9 hours ago	Ready
ip-10-0-3-185.us-east-2.compute.internal	t3.medium	node-group-1-20230730183801533800000013	Created 9 hours ago	Ready

Node groups (1) [Info](#)


EditDeleteAdd node group

Group name	Desired size	AMI release version	Launch template	Status
node-group-1-20230730183801533800000013	2	1.27.3-20230711	one-20230730183800838500000011 (1)	Active

AWS VMs

 Services [Option+S]

VPC EC2 IAM CloudFormation Elastic Kubernetes Service S3

 New EC2 Experience
Tell us what you think

EC2 Dashboard

EC2 Global View

Events







▼ Instances

Instances

Instance Types

Instances (4) [Info](#)

Instance state = running

<input type="checkbox"/>	Name ▾	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status	Availability Zone
<input type="checkbox"/>	Jenkins	i-0385d48bc7b8684a1	✓ Running  	t2.medium	✓ 2/2 checks passed	No alarms +	us-east-2a
<input type="checkbox"/>	node-group-1	i-0e3ee17d5baf82e6c	✓ Running  	t3.medium	✓ 2/2 checks passed	No alarms +	us-east-2b
<input type="checkbox"/>	node-group-1	i-0f25d5c4e9a5c752c	✓ Running  	t3.medium	✓ 2/2 checks passed	No alarms +	us-east-2a

Instance state ▾


k9s

~/ssh

~zsh...

k9s

Context: arn:aws:eks:us-east-2:676528066169:cluster/veronafinal-Jo9Qo7th
Cluster: arn:aws:eks:us-east-2:676528066169:cluster/veronafinal-Jo9Qo7th
User: arn:aws:eks:us-east-2:676528066169:cluster/veronafinal-Jo9Qo7th
K9s Rev: v0.27.4
K8s Rev: v1.27.3-eks-a5565ad
CPU: n/a
MEM: n/a



Pods(kube-system)[12]

NAME↑	PF	READY	RESTARTS	STATUS	IP	NODE
aws-node-7nv98	●	1/1	0	Running	10.0.2.41	ip-10-0-2-41.us-east-2.
aws-node-jj97v	●	1/1	0	Running	10.0.1.69	ip-10-0-1-69.us-east-2.
coredns-647484dc8b-dv98d	●	1/1	0	Running	10.0.1.121	ip-10-0-1-69.us-east-2.
coredns-647484dc8b-15msb	●	1/1	0	Running	10.0.2.247	ip-10-0-2-41.us-east-2.
ebs-csi-controller-65ff9665ff-4q7v2	●	6/6	0	Running	10.0.1.110	ip-10-0-1-69.us-east-2.
ebs-csi-controller-65ff9665ff-mhz1f	●	6/6	0	Running	10.0.2.114	ip-10-0-2-41.us-east-2.
ebs-csi-node-vqwgr	●	3/3	0	Running	10.0.1.210	ip-10-0-1-69.us-east-2.
ebs-csi-node-xc7km	●	3/3	0	Running	10.0.2.98	ip-10-0-2-41.us-east-2.
fluentd-75b41	●	1/1	0	Running	10.0.2.37	ip-10-0-2-41.us-east-2.
fluentd-jcc9g	●	1/1	0	Running	10.0.1.230	ip-10-0-1-69.us-east-2.
kube-proxy-6vfjm	●	1/1	0	Running	10.0.1.69	ip-10-0-1-69.us-east-2.
kube-proxy-rfrdg	●	1/1	0	Running	10.0.2.41	ip-10-0-2-41.us-east-2.

<namespace>

<pod>












CI/CD

Use CI/CD tools - e.g. Jenkins - to build and automate software and application code pipelines

1. Build pipelines with CI/CD configuration files - e.g. Jenkinsfile and declarative pipeline syntax
2. Install and manage CI/CD plugins where applicable - e.g. Jenkins plugins such as Docker plugin
3. Knowledge of alternative CI/CD tools other than the one used and understand the similarities and differences - e.g. GitHub Actions
4. Successfully deploy an app through multiple development stages and environments using CI/CD - e.g. dev, staging, prod and pulling source code from branches such as main

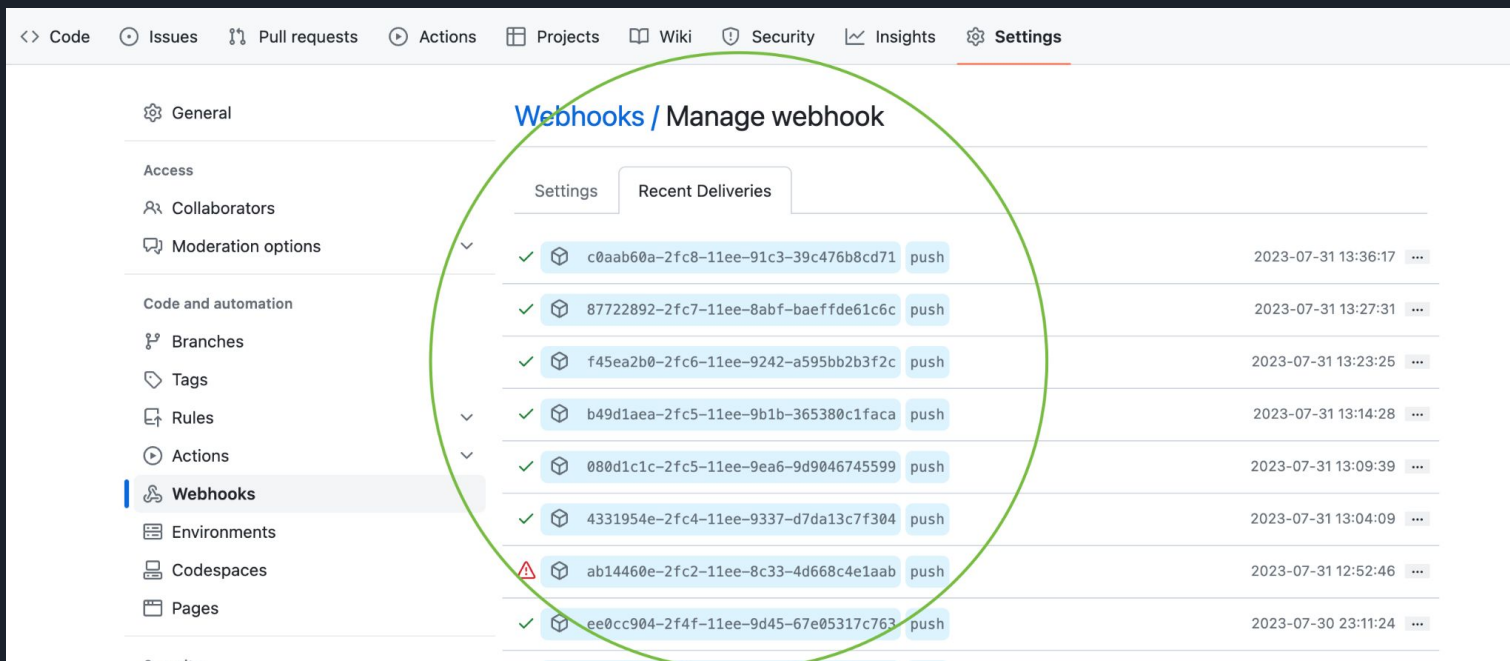
Jenkins Tokens for Github and plugins

Credentials

T	P	Store ↓	Domain	ID	Name
		System	(global)	18c03192-4ed6-481d-806f-c3956496a262	18c03192-4ed6-481d-806f-c3956496a262
		System	(global)	github-access	
		System	(global)	github	
		System	(global)	dockerhub	dockerhub's token
		System	(global)	aws-iam	AKIAZ3BBXBZ432QVR7TC (aws-iam-verona2)

Github & Jenkins

Github webhook for triggering Jenkins CD process when commits



The screenshot shows the GitHub 'Settings' page for a repository. The 'Webhooks' section is selected in the left sidebar. The 'Recent Deliveries' tab is active, showing a list of webhook deliveries. A green circle highlights this tab and the list of deliveries.

Settings	Recent Deliveries
✓	<div>✓ c0aab60a-2fc8-11ee-91c3-39c476b8cd71 push</div> <div>2023-07-31 13:36:17 ...</div>
✓	<div>✓ 87722892-2fc7-11ee-8abf-baeffe61c6c push</div> <div>2023-07-31 13:27:31 ...</div>
✓	<div>✓ f45ea2b0-2fc6-11ee-9242-a595bb2b3f2c push</div> <div>2023-07-31 13:23:25 ...</div>
✓	<div>✓ b49d1aea-2fc5-11ee-9b1b-365380c1faca push</div> <div>2023-07-31 13:14:28 ...</div>
✓	<div>✓ 080d1c1c-2fc5-11ee-9ea6-9d9046745599 push</div> <div>2023-07-31 13:09:39 ...</div>
✓	<div>✓ 4331954e-2fc4-11ee-9337-d7da13c7f304 push</div> <div>2023-07-31 13:04:09 ...</div>
⚠	<div>⚠ ab14460e-2fc2-11ee-8c33-4d668c4e1aab push</div> <div>2023-07-31 12:52:46 ...</div>
✓	<div>✓ ee0cc904-2f4f-11ee-9d45-67e05317c763 push</div> <div>2023-07-30 23:11:24 ...</div>

Github & Jenkins & Dockerhub repo

Build docker image and push to docker repo using Jenkins

```
pipeline {
    agent any

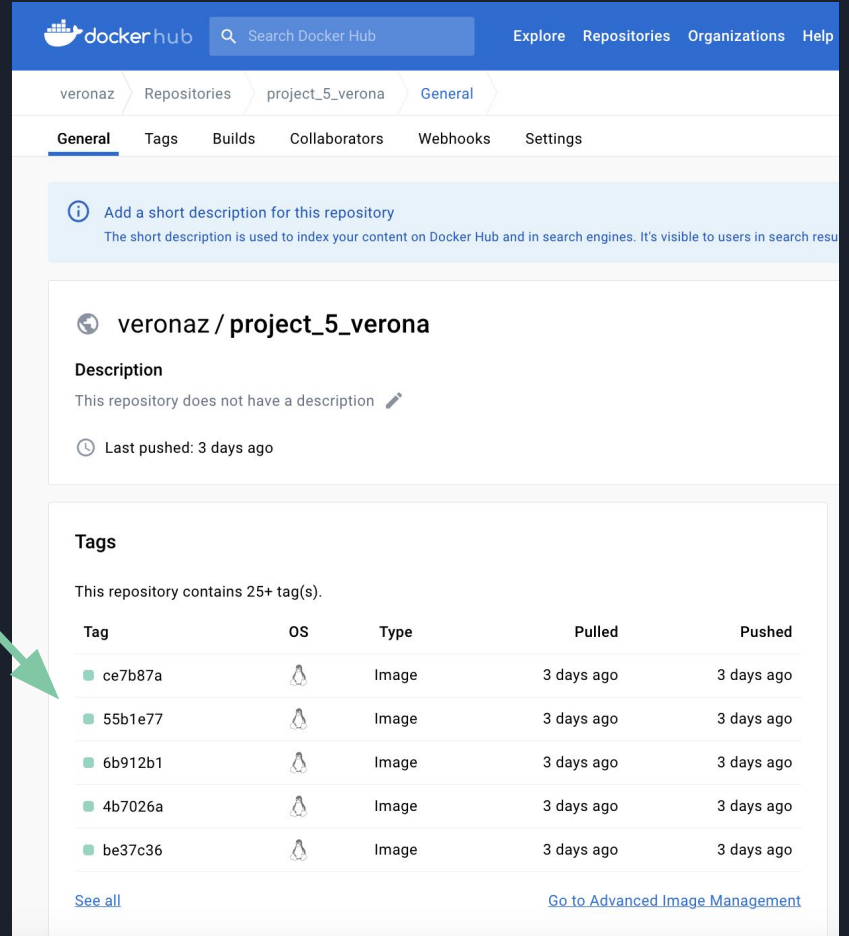
    stages {
        stage('Build') {
            steps {
                script {
                    dir ("React-application") {
                        sh("echo Get current git hash")
                        sh("printenv")
                        gitHash = sh(script: "git rev-parse --short HEAD | tr -d '\n'", returnStdout: true)

                        sh("docker build --target runtime . -t veronaz/project_5_verona:${gitHash}")
                        sh("docker build --target test . -t veronaz/project_5_verona_test:${gitHash}")
                    }
                }
            }
        }
    }
}
```

Dockerhub repo

With Git commit hash

\$githash



The screenshot shows the Docker Hub interface for the repository 'veronaz / project_5_verona'. The 'General' tab is selected, and the 'Tags' section is visible. A green arrow points from the '\$githash' text in the previous block to the first tag, 'ce7b87a'.

veronaz / project_5_verona

Description
This repository does not have a description

Last pushed: 3 days ago

Tags
This repository contains 25+ tag(s).

Tag	OS	Type	Pulled	Pushed
ce7b87a	linux	Image	3 days ago	3 days ago
55b1e77	linux	Image	3 days ago	3 days ago
6b912b1	linux	Image	3 days ago	3 days ago
4b7026a	linux	Image	3 days ago	3 days ago
be37c36	linux	Image	3 days ago	3 days ago

[See all](#) [Go to Advanced Image Management](#)

Jenkinsfile

The screenshot shows a VS Code editor interface with a dark theme. On the left, the Explorer sidebar displays a project structure for 'DEVOPSFINALCAPSTONPROJECT'. The file 'Jenkinsfile' is selected and highlighted in blue. The main editor area shows the content of the Jenkinsfile, which is a YAML script for a Jenkins pipeline. The pipeline includes steps for uploading a Docker image to Docker Hub and deploying it to an AWS EKS cluster. The bottom of the editor shows the 'TERMINAL' tab, which contains the output of a git push command.

EXPLORER

- DEVOPSFINALCAPSTONPROJECT
 - data.tf
 - eks.tf
 - fluentd.tf
 - jenkins.tf
 - main.tf
 - outputs.tf
 - prometheus_helm.tf
 - README.md
 - terraform.tf
 - terraform.tfstate
 - terraform.tfstate.backup
 - variables.tf
 - vpc.tf
 - React-application
 - kubernetes_manifest
 - deployment.yaml
 - fluentd-config-map.yaml
 - fluentd-dapr-with-rbac.yaml
 - service.yaml
 - public
 - scripts
 - src
 - .gitignore
 - docker-compose.yml
 - Dockerfile
 - Dockerfile.dev
 - Jenkinsfile**
 - package-lock.json
 - package.json

Jenkinsfile

```
29     steps {
30         script {
31             echo 'Uploading docker image ...'
32             withCredentials([usernamePassword(credentialsId: 'dockerhub', passwordVariable: 'dockerHubPassword')]) {
33                 sh("docker login -u ${env.dockerHubUser} -p ${env.dockerHubPassword}")
34                 sh("docker push veronaz/project_5_verona:${gitHash}")
35             }
36             echo 'Deploying....'
37             echo 'Updating deployment manifest image tag...'
38             dir ("React-application") {
39                 sh("sed -i \"s,IMAGE_TAG,${gitHash},g\" kubernetes_manifest/deployment.yaml")
40                 withAWS(credentials: 'aws-iam', region: 'us-east-2') {
41                     script {
42                         sh ("aws eks update-kubeconfig --name veronafinal-Jo9Qo7th --region us-east-2")
43                         sh ("kubectl create namespace verona --dry-run -o yaml | kubectl apply -f -")
44                         sh ("kubectl apply -f ./kubernetes_manifest")
45                         sh ("kubectl get -f ./kubernetes_manifest/service.yaml")
46                     }
47                 }
48             }
49         }
50     }
```

TERMINAL

```
After doing this, you may fix the identity used for this commit with:

git commit --amend --reset-author

1 file changed, 1 insertion(+), 1 deletion(-)
● verona@veronas-MacBook-Air DevopsFinalCapstonProject % git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9) done
```

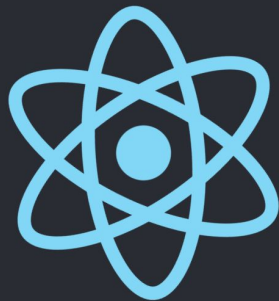


Jenkinsfile

Kubernetes (aws eks)

```
script {  
    sh ("aws eks update-kubeconfig --name veronafinal-Jo9Qo7th --region us-east-1")  
    sh ("kubectl create namespace verona --dry-run -o yaml | kubectl apply -f -")  
    sh ("kubectl apply -f ./kubernetes_manifest")  
    sh ("kubectl get -f ./kubernetes_manifest/service.yaml")  
}
```

Webapp deployment demo



Hi there, this is Verona!

[Learn React](#)

ChromeFileEditViewHistoryBookmarksProfilesTabWindowHelp

zoom

Mon Jul 31 1:34 PM

Not Secure3.16.252.160:8080

Jenkins

Search (⌘+K)

verona

log out

Dashboard

New Item

People

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

All

+

S	W	Name ↓	Last Success	Last Failure	Last Duration
		FinalProject	6 min 56 sec #10	24 min #7	27 sec
		test	18 hr #1	N/A	64 ms

Icon: S M L

Icon legend

Atom feed for all

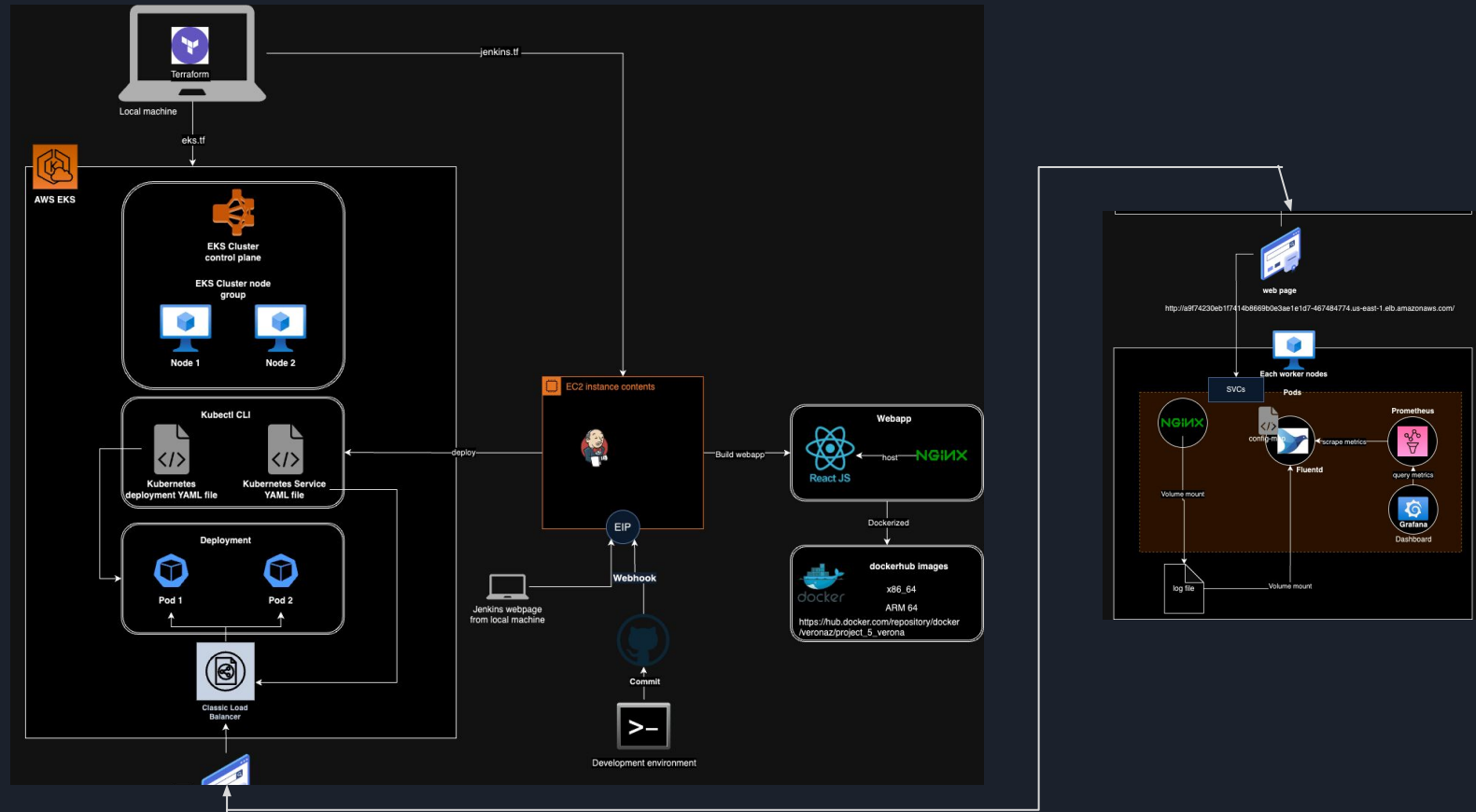
Atom feed for failures

Atom feed for just latest builds

REST API

Jenkins 2.401.3

Architectural Diagram



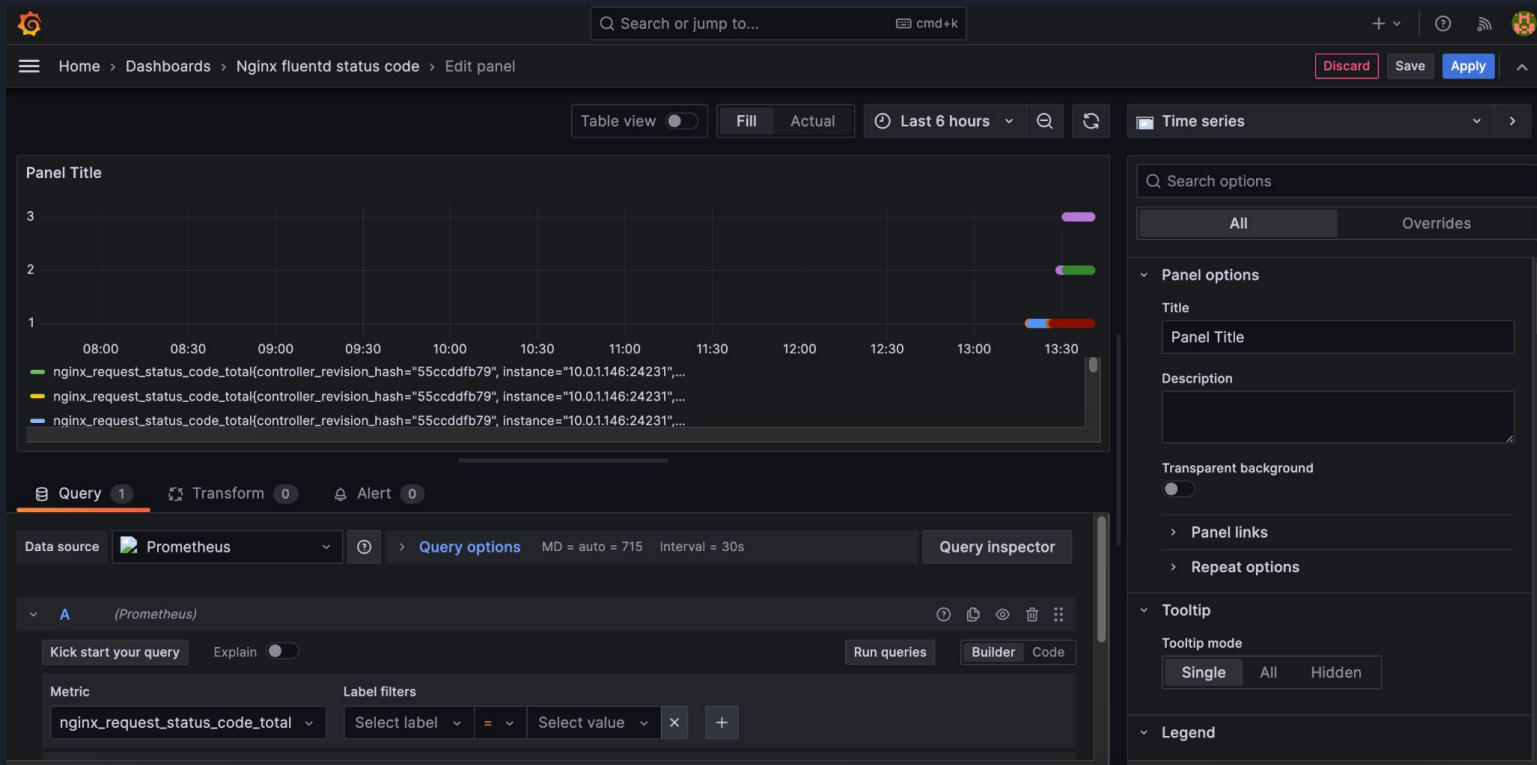
Observability systems

Set up monitoring and logging systems

e.g. FluentD, ELK, Prometheus

1. Leverage UI/dashboard tools to monitor the infrastructure & app status - e.g. Grafana

Fluentd(nginx), prometheus, grafana



Grafana & Prometheus

Added data source (prometheus) in Grafana console

The screenshot displays the Grafana web interface. The top navigation bar includes a search bar with the text "Search or jump to...", a command prompt "cmd+k", and several utility icons. The breadcrumb trail indicates the current location: "Home > Administration > Data sources > Prometheus".

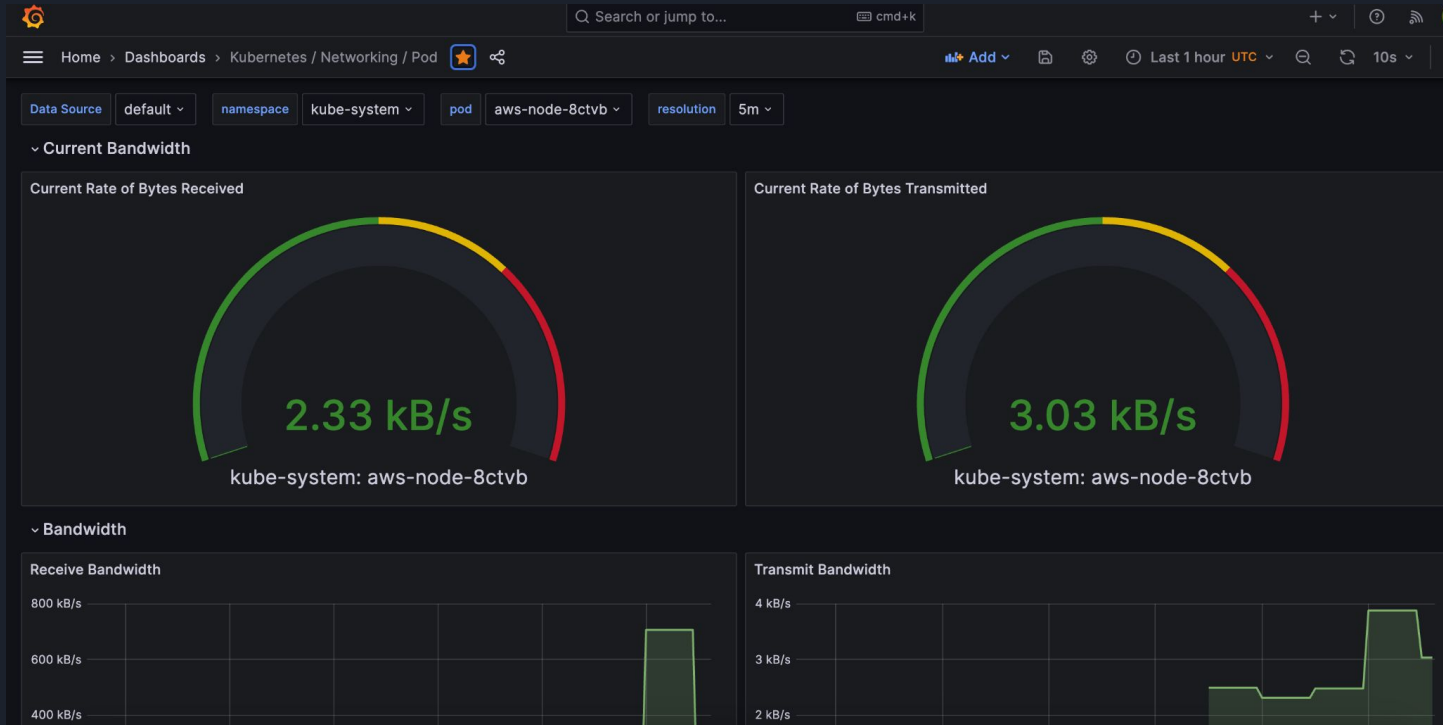
On the left sidebar, the "Administration" menu is expanded, showing options like "Data sources", "Plugins", "Users", "Teams", "Service accounts", "Default preferences", "Settings", "Organizations", and "Stats and license".

The main content area is titled "Prometheus" and shows "Type: Prometheus". It has two tabs: "Settings" (active) and "Dashboards". A blue informational box states: "Provisioned data source. This data source was added by config and cannot be modified using the UI. Please contact your server admin to update this data source." Below this, a green badge indicates "Alerting supported".

The configuration section shows the "Name" as "Prometheus" and its status as "Default" with a toggle switch. Under the "HTTP" section, there are three rows of configuration:

Field	Value
Prometheus server URL	http://prometheus-kube-prometheus-prom...
Allowed cookies	New tag (enter key to add) [Add]
Timeout	Timeout in seconds

Grafana Dashboard





Appendix - Final presentation

Presentation of the DevOps project 15%

1. Architecture diagrams of project components
2. Clearly communicate and articulate project goals, objectives and results to faculty and peers
3. Describe limitations of project design and make recommend recommendations for improvements
4. Ability to lead and answer Q&A session