Devops combination of cultural philosophies, practices and tasks.

Infrastructure As Code → Terraform: Infrastructure Provisioning
→ Ansible: configration management

A COMPUTER NETWORK COMPRISES TWO OR MORE COMPUTERS THAT ARE CONNECTED - either by cables or Wifi WITH THE PURPOSE OF TRASMITTING, SHARING DATA AND RESOURCES.

LAN (local area network) short distance
WAN (wide are network) wide area - region to region
VPN (virtual private net) secure, point to point connection
IP Adress: unique number assigned to every device connected to net.
NODES: a connection point → printers, modem, switches.
ROUTERS: physical or virtual device that sends information contained in data packets between networks.
SWITCHES: a device that connects other devices and manages node-to-node communication within a network, ensuring data packets reach their ultimate destination.
PORT → identifies a specific connection between network devices. Numbers

NETWORK TOPOLOGY
BUS NET TOPOLOGY → Every network node is directly connected main cable.
RING TOPOLOGY → nodes are connected in a loop
STAR NETWORK TOPOLOGY → all nodes connected to a single, central hub
MESH TOPOLOGY → overlapping connections between nodes
LOAD BALANCERS → efficiently distribute tasks, workloads, and network traffic across available servers. AIR TRAFFIC CONT (A)

VIRTUALIZATION: is the process of using special software on a physical machine - to create virtual machines.
Special software is called "hypervisor"

Container technology, also simply known as just a container, is a method to package an application so it can be run, with its dependencies, isolated from other processes.

VIRTUALIZATION ENABLES YOU TO RUN MULTIPLE OPERATING SYS. ON THE HARDWARE OF A SINGLE PHYSICAL SERVER, WHILE CONTAINERIZATION ENABLES YOU TO DEPLOY MULTIPLE APPLICATIONS USING THE SAME OS ON A SINGE VIRTUAL MACHIENE

VM → SIZE Large takes minutes run
CONTAINER → SIZE LIGHT takes seconds run

```
k config set-context --namespace=dev
k run test --image=redis → k get po
k exec -it test -- /bin/bash
k create deploy fron-end --image=nginx
k expose deploy front-end --name=front-end-svc --port=80 --type=NodePort
k scale --replicas=3 deploy mysql
```

kubelet = a container runtime → start-stop container
kube-proxy = network - loadbalancing - svc
scheduler: which pod goes where
Daemonset: ensures that all nodes run a copy of a Pod.
container runtime → responsible for running containers.

```
FROM node:14                 Pulling offical node image From Docker
WORKDIR /usr/src/app         SETTING WORKING DIR. APP
COPY package*.json ./        COPY Package.json file to Work D.
RUN npm install              Install needed packages
COPY . .                     COPY code files to Work Dir.
EXPOSE 5001                  → Expose publicly accessible
CMD ["node", "index.js"]     RUN index.js when container launches.
```

KUBERNETES	DOCKER SWARM
Auto-Scaling	No - Auto Scaling
Difficult to Start Cluster	Easy to start cluster
Desired State Management	

Auto Scaling: adjust amount of computational resources.

NO DEPOSIT NO RETURN

. LOG IN POD and PING THE SVC

IN THIS ROLE, I WILL ALWAYS COLLABORATE with my co-workers to ensure we quickly meet the team's objectives.

```
SELECT * FROM student_table WHERE grade < 90
SELECT * FROM employees ORDER BY gender DESC, first_name ASC;
```

<p> My cat is very grumpy </p>
Opening Tag — content — Element — Closing Tag

CSS - Cascading Style Sheets
<p style="color: blue;"> Hello World </p>

ANSIBLE WORKS AGAINST MULTIBLE MANAGED NODES OR "HOSTS" IN YOUR INFRASTRUCTURE AT THE SAME TIME, USING A LISTS KNOWN AS INVENTORY (STATIC, DYNAMIC)
PLAYBOOKS CONTAINS PLAYS
PLAYS CONTAINS TASKS
TASKS CALL MODULE
TASK RUN SEQUENTIALLY

ANSIBLE → Simplest, No Support Windows
PUPPET → COMPLEX, SUPPORT ALL OS

PROMETHEUS → SYSTEM MONITORING AND ALERTING TOOLKIT.
GRAFANA → VISUALIZATION - ANALYTICS SOFTWARE Beatiful graph.
GATHER OPERATIONAL METRICS, RAISE ALERT

AGILE → DIVIDING PRODUCT DEVELOPMENT LIFE CYCLE INTO SMALL COMPONENTS
SCRUM → Sprints 2 week, Releases faster x1
KANBAN → CONTINOUS - Releases X2.5

POSTGRESQL → object based Relational DB supports pyhton and .net
used case → most complex websites, app
MYSQL → SIMPLE SOLUTIONS - SPEED

HELM uses charts to pack all the required K8s components for an application to deploy run and scale. A chart root has to have only one file named chart.yml by convention
Upload HELM chart in S3 bucket
Terraform → Hashicorp Configuration Lang.
GITOps → Treat Infrastructure Code the same as application code.
Iac - Version control - Pull Merge requests - CI/CD Pipeline.

Jenkins is an open source CI tool which is written in Java. It keeps a track on version control system to initiate and monitor a build system if any changes occur. It monitors the whole process and provides reports and notifications to alert the concern team.

- IAM user access issue → EC2 volumes encryted with KMS to give required permission.
- EKS - worked nod has internet access but pod has not internet access why? Tagging (labelling) is really important!
  kubectl exec -it podname -- sh
  curl google.com → no connection why?
  check documentation → you need to add tags to private subnets.
- Deployment failed. (failed to assign IP container) default settings of CNI gives less IP for performance I guess. You can set env MINIMUM_IP_TARGET. EKS Private. Monitor with Promethus. keep CIDR large

DNS system works much like a phone book by managing the mapping between names and numbers.

Messep Up → Billing Dashbord Notification

STATELESS → does not keep track → HTTP Web Search
STATEFUL → keep tracks, DB - Banking.

AS FAR AS DEPLOYMENT IS CONCERNED, PODS ARE INTERCHANGEABLE. WHILE A STATEFUL KEEPS A UNIQUE IDENTITY FOR EACH POD IT MANAGES.

SENSE OF OWNERSHIP
NEVER STOP LEARNING

JIRA → is a tool develop to help teams for project management, bug tracking, an issue tracking. IT IS A ISSUE TRACKER. → JIRA SOFTWARE → SCRUM MASTER creates and manages sprints
TO DO | In-Progress | Done

- Positive = SMILE ☺
- self sufficient
- go beyond expectation

HOW DO YOU HELP YOUR TEAM GROW PROFESSIONALLY?

RANCHER IS CLUSTER MANAGEMENT TOOL
YOU CAN USE RANCHER WITH DIFF CLOUD PROV.

HELM IS PACKAGE MANAGER FOR K8s
HELM is a collection of all your versioned pre-configured application resources which can be deployed as one unit. You can then deploy another version of chart with a different set of configurations

KOMPOSE is a conversion tool, for Docker Compose to container orchestrators such as K8s - Opensh.

```
kompose convert -f dockercompose.yml -o k8s/base
kubectl apply -f .
kubectl get po
```

KUSTOMIZE configuration management solution declarative yaml artifacts (patches) that selectively override default settings without actually changing the original files.

MAVEN is a tool that can be used for building and managing any Java based project.

JENKINS PIPELINE
- Create ECR repo
- Packaging the app into jars with maven
- Preparing Tags for Docker images
- Build app Docker images
- Push images to ECR repo
- Create key pair for Ansible
- Create Docker Swarm with CFN-Terraform
- Deploy App on Docker Swarm → Ansible
- Test the Application Deployment
- Run QA Automation Test, Functional?
- Tear down docker swarm

DOCKER as a container engine makes it easy to take a piece of code & run it inside a container. (Dockerize for ease of shipping deployment) We add all the build instructions, such as the code dependencies, commands, and base image details, in Dockerfile.
Also, with container orchestration tools like K8s, it is best to have small-sized images to reduce the image transfer and deploy time.

CLOUDWATCH
GITOPS Treat Infrastructure Code the same as Aplication Code.

CONTAINERS are deployed applications bundled with all necessary dependencies and config files
VIRTUALIZATION is an abstract version of a physical machine, while CONTAINERAZATION is the abstract version of application.

CONTAINERS' LIFE CYCLE: create container, run container, pause, unpause, start, stop
- Browser Issue, check logs, check servers up? memory, cpu → metrics, system load metrics, node.js, java → garbage collection
- Build and Deploy Issue → CI/CD Pipeline, test, build, deploy run command manually

CLUSTER IP → Reachable within cluster. Expose service on a cluster-internal IP
NODE PORT → Expose node service at a static port
LOAD BALANCER → Expose service cloud providers load balance.

A sofware intermediary that allows two application to talk each other.
API stands for application programming interface which is a set of definitions and protocols for building and integrating app software
SLO → Service Level Objectives

K8s → livenes → whether container is running?
→ readiness → whether container ready to receive Request

THINGS BACKUP
JENKINS → CI tool, keep track of version Cont. Sy. monitor the build system, provide notifications, alert.
JENKINS PLUGINS → Maven 2 project, Amazon EC2, HTML Publisher, Git Plugin, Metrics, GIT, Jococo
SCHEDULE JENKINS → Trigger Webhook SCM, Manual request Build, CRON jobs.

CRITICAL EVENT MANAGEMENT PROVISION.

Saying that an operation is idempotent means that applying it multiple times will not change the result.

SMTP Simple Mail Transfer Protocol, and it's an application used by mail servers to send, receive, and/or relay outgoing mail between email senders and receivers.

TCP/IP Transmission Control Protocol/Internet Protocol is a set of network Protocol → that enable communication between computers

SSH Secure Shell is a network communication protocol that enables two computers to communicate and share data.

TERRAFORM module is a set of Terraform configuration files in a single directory.

CURL will show the output in the console.
WGET will download it into a file.
MAVEN is built automation tool for Java projects.
ITIL → best practices for delivering IT services.

```
- cat /etc/*release*
- docker rmi ubuntu (remove image also)
```

Jenkins is a web server

docker run -e APP_COLOR=blue image red

```
FROM ubuntu
ENTRYPOINT ["sleep"]   → docker run ubuntu-sl 10
CMD ["5"]              → You can override on CLI
```

Otherwise it will sleep 5 seconds.

ROLE-BASED ACCESS CONTROL (RBAC)
RBAC restricts network access based on person's role within an organization and has become one of main methods for advanced access control.
Employees are only allowed to access the information necessary to effectively perform their job duties.

API TOKEN is a unique identifier of an application requesting access to your service.

A Taint is a property applied to a node to enable it to run only specific Pods. Only a pod which has a toleration to the specific taint is capable of being scheduled on the "tainted" node.
A Toleration is a property added to a Pod which enables it to run on a specific tainted nodes
NodeAffinity is added to Pod to make sure Pod sch sp. nodes.

ci job	dev feature bugfix	Webhook on each commit	Unit Test
nightly	dev	every night 11:59 pm	Functional T.
weekly	release	every sunday 11:59 pm	Manual QA
weekly staging	release	every sunday 11:59 pm	User Accepting Testing Staging
prod	main	webhook each commit	Production Env.

This project aims to create full CI/CD Pipeline for microservice based applications.

- ./mvnw clean package ⇒ Package the application with maven wrapper.
- Kompose convert -f K8s/docker-compose-yaml -o K8s/base
- * Init Containers: oluşturduğumuzda içinde bulunduğu containerın başlamasını bir başka (init container) bağlamış oluyoruz. Örnekte Config server var. Conf.Ser. bir sınırlama yok ama discovery serverın Spec kısmına init Container koyuyuz ve diyoruz ki Config başlamadan Discovery server başlatma!

- Packing the app into jars with maven
- Preparing tags for Staging (prod) Docker images
- Build App Docker Images
- Push Images ECR Repo
- Deploy App on K8s Cluster
  - sh rancher login
  - sh envsubst kust.temp.ym -kustomization.ym
  - sh rancher kubectl create secret..
  - sh rancher kubectl apply -k K8s/staging
- post {
  always {
  echo "Deleting all local images."
  sh 'docker image prune -af'
  }
  }

CI Job. when developer push codes to (dev, feature, bug) webhook triggers Jenkins job that starts, build and unit test. jenkins - maven - git - github - Jacoco (Java code coverage tool) that tests source code show code coverage rate

- Create ECR repo
- Packing the app into jars with maven
- Preparing Tags for Docker Images
- Build App Docker Images
- Push to images ECR Repo
- Create Key Pair for Ansible
- Creating QA Automation Infrastructure for Dev Environmet with Cloud-formation
- Setup Docker Swarm for QA Automation Build
  - Update dynamic enviroment
  - swarm setup for all nodes (instances) (install docker aws cli)
  - Swarm setup for Grand Master node → init swarm
  - Swarm setup for mangers → Join docker manager
  - Swarm setup for worker → Join docker worker
- Deploy App on Docker Swarm (Ansible Playbook)
- Test the Application Deployment (App ready or not?)
  - sh "curl -s $(terraform output -raw GRAND_MASTER_PUBLIC_IP):8080"
- Run the Selenium Functional Test on QA Environment
- TEAR DOWN
  - Deleting all local images 'docker image prune -af'
  - Delete the Image Repo on ECR aws ecr delete-repository
  - Tear down docker swarm infrastructure 'aws cloudformation delete-stack --region....'

DOCKER

If we have more than one CMD command in Dockerfile it will only consider the latest one.
We can overwrite CMD file but can we ENTRYPOINT ENTRYPOINT will not overwrite it will APPEND. we can use CMD and ENTRYPOINT together.

```
FROM Ubuntu                    docker run -it imagename "Hello A"
ENTRYPOINT ["echo"]
CMD ["Hello World"]            * we can overwrite CMD on CLI
```

Two types of Volume mounting →
Volume Mounts → within Volume directory default → /var/lib/docker/volumes
Bind mounts → Any directory on docker host.
docker run --mount type=bind,source=data,mysql target=/var/lib/mysql

REDUCE DOCKER IMAGE SIZE

1. Use minimal Base images or distroless images
2. Use multistage builds
3. Minimize the number of layers
4. Understanding Caching → rebuilt again again stores caching. (COPY after FROM)
5. Use Dockerignore
6. Keep Application Data Elsewhere (will decrease size of images)

It is highly recommended to use the volume feature of the container runtimes to keep the image seperate from the data.

```
FROM node
ENV MONGO_DB_USERNAME=admin MONGO_DB_PASSW
RUN mkdir -p /home/app
COPY . /home/app
CMD ["node", "server.js"]
```

JENKINS Freestyle Project (create bash or shell commands)
Pipeline : Workflow written in DSL.

CI) You take the code and packaging up and giving the CI process. CI process make sure the code is passed tests.

CD) Deploy the code some system like VM, container

Continious Delivery has some manual intervention.
Continious Deployment is automatically deploy after CI (dept) There is no manual interaction.

K8s → What are the two diffent ways inject pod? (use env vars, configmaps, secrets)
There are other better ways of handling sensitive data in K8s, such as using Helm secrets, Hashicorp Vault
You can specify the amount of cpu and memory a container needs.
- If a container tries to use more CPU than its limits it will THROTTLE the CPU. The container can not use more CPU than its limits.
- If a pod tries to use more memory than its limit constantly the pod will be terminated.

STATEFULSETS Like a deployment a SS manages Pods that are based on an identical spec. Unlike a Deployment, a SS maintains a sticky identity for each of their Pods. These pods are created from the same spec. but are not interchangeable, each has a persistent identifier that it maintains across any rescheduling.
Statefulset ensures stable storage for pods.

ANSIBLE Simple, Powerful, Agentless

Ansible is agentless meaning that you do not need to install any software to target machine
- ANSIBLE needs simple SSH connections.
- Inventory file location ⇒ /etc/ansible/hosts
- roles

PLAYBOOK - A single YAML file
PLAY ⇒ Defines a set of activities (tasks) to be run on hosts.
TASK ⇒ An action to be performed on the host
- Execute a command, Run a Script
- Install a package, Shutdown/Restart

« ansible-playbook playbook.yaml »

Idempotency ⇒ An operation is idempotent if the result of performing it once is exactly the same as the result of performing it repeatedly without any intervening actions.

ROLES → main point is re-usable ANSIBLE-GALAXY
« ansible-galaxy init mysql »
Roles let you automatically load related vars, files, tasks, handlers and other ansible artifacts.
HANDLERS → Sometimes you want a task to run only when a change is made on a machine. For example, you may want to restart a service if a task updates the configuration of that service, but not if configuration is unchanged.
HELM is a package manager. We use a single command to install entire application. « helm install wordpress »
CHARTS are collection of files, and they contain all the instructions, that helm needs to be able to create collection of objects that you need in your K8s cluster

```
helm repo
helm repo list
helm repo update
```

How IAM Works
Users, Groups, Roles and Policies / Permissions
specific individuals or users | collection of users | collection of policies | Allow / Deny Resources

- Use least privilege Model
- Exercise caution when modifying policies.

GIT
git config user.name "fatihtepe"
git config user.email "fatihtepe@gmail.com"
git checkout -b testbranch ⇒ checkout -create
git branch testbranch → create branch
git checkout testbranch → switch to testbranch
git merge ⇒ keeps both master and feature branch history
git rebase -i HEAD~4 ⇒ interactively rebase last 4 commits — pick — squash — squash
pick ⇒ delete ⇒ write squash changes will merge to pick. 4 commits combines to one
git revert ⇒ you undo changes and keep the record of that change in history.
git reset --hard HEAD~1 ⇒ take you previous commit without saving changes
git stash → take changes to another area later you will commit
git reflog → shows all logs even git reset

PROMETHEUS - The most famous metrics exporter. It collects and stores metrics. You can set up alerting rules also
GRAFANA visualizes metrics collected by Prometheus in pretty dashboard. Grafana is a universal tool and it can visualize data from many sources, including Prometheus SQL databases.
You can set up alerting rules in Grafana Also!

If you want to hear more about my experience I will be glad to answer your questions.