**Activités Types et Compétences Professionnelles**

**1. Automatiser la création de serveurs à l’aide de scripts**

**Explication**: Cette compétence consiste à utiliser des scripts pour créer et configurer des serveurs automatiquement, sans intervention manuelle. Cela permet de gagner du temps et d'assurer une configuration uniforme et reproductible.

**Exemple**: Utiliser un script Shell (Bash) pour installer Apache et configurer un serveur web.

bash

Copier le code

#!/bin/bash

# Mise à jour des paquets

sudo apt-get update

# Installation d'Apache

sudo apt-get install -y apache2

# Démarrage du service Apache

sudo systemctl start apache2

# Activer Apache pour qu'il démarre au démarrage

sudo systemctl enable apache2

echo "Serveur Apache installé et démarré."

**2. Automatiser le déploiement d'une infrastructure**

**Explication**: Cela implique l'utilisation d'outils d'infrastructure as code (IaC) pour déployer et gérer les infrastructures. Cela permet de définir l'infrastructure en tant que code, ce qui facilite la gestion des versions et le déploiement automatisé.

**Exemple**: Utiliser Terraform pour déployer une infrastructure sur AWS.

hcl

Copier le code

provider "aws" {

region = "us-west-2"

}

resource "aws\_instance" "example" {

ami = "ami-0c55b159cbfafe1f0"

instance\_type = "t2.micro"

tags = {

Name = "example-instance"

}

}

**3. Sécuriser l’infrastructure**

**Explication**: Cette compétence consiste à mettre en place des mesures de sécurité pour protéger l'infrastructure contre les menaces. Cela inclut la configuration de pare-feu, la gestion des accès, la surveillance de la sécurité, et la mise en œuvre des bonnes pratiques de sécurité.

**Exemple**: Utiliser Ansible pour configurer un pare-feu sur un serveur.

yaml

Copier le code

---

- name: Configurer un pare-feu avec UFW

hosts: all

become: yes

tasks:

- name: Installer UFW

apt:

name: ufw

state: present

- name: Autoriser les connexions SSH

ufw:

rule: allow

port: 22

- name: Autoriser les connexions HTTP

ufw:

rule: allow

port: 80

- name: Activer UFW

ufw:

state: enabled

**4. Mettre l’infrastructure en production dans le cloud**

**Explication**: Cela consiste à déployer et gérer une infrastructure en production dans un environnement cloud, en utilisant les services et outils disponibles sur les plateformes cloud comme AWS, Azure, ou Google Cloud.

**Exemple**: Utiliser AWS CloudFormation pour déployer une pile d'infrastructure en production.

yaml

Copier le code

AWSTemplateFormatVersion: '2010-09-09'

Resources:

MyEC2Instance:

Type: 'AWS::EC2::Instance'

Properties:

InstanceType: 't2.micro'

ImageId: 'ami-0c55b159cbfafe1f0'

KeyName: 'my-key-pair'

SecurityGroups:

- !Ref MySecurityGroup

MySecurityGroup:

Type: 'AWS::EC2::SecurityGroup'

Properties:

GroupDescription: 'Enable SSH and HTTP access'

SecurityGroupIngress:

- IpProtocol: 'tcp'

FromPort: '22'

ToPort: '22'

CidrIp: '0.0.0.0/0'

- IpProtocol: 'tcp'

FromPort: '80'

ToPort: '80'

CidrIp: '0.0.0.0/0'

**Utilisation de ces compétences dans une formation DevOps**

**1. Automatiser la création de serveurs à l’aide de scripts**

**Formation**: Les étudiants apprennent à écrire des scripts Bash pour automatiser l'installation et la configuration de logiciels sur des serveurs. Par exemple, ils créent des scripts pour installer des serveurs web, des bases de données, ou des environnements de développement.

**2. Automatiser le déploiement d'une infrastructure**

**Formation**: Les étudiants utilisent des outils comme Terraform pour déployer des infrastructures complètes, y compris les réseaux, les instances de calcul, et les services de stockage. Ils apprennent à écrire des fichiers de configuration et à exécuter des commandes pour créer et gérer l'infrastructure de manière programmatique.

**3. Sécuriser l’infrastructure**

**Formation**: Les étudiants apprennent les meilleures pratiques de sécurité et utilisent des outils comme Ansible pour automatiser la configuration de la sécurité. Ils configurent des pare-feu, gèrent les accès, et mettent en place des systèmes de surveillance pour garantir la sécurité de l'infrastructure.

**4. Mettre l’infrastructure en production dans le cloud**

**Formation**: Les étudiants déploient des applications en production sur des plateformes cloud comme AWS, Azure, ou Google Cloud. Ils utilisent des outils natifs des fournisseurs de cloud pour gérer le déploiement, la mise à l'échelle, et la surveillance de leurs applications en production.

En intégrant ces compétences dans une formation DevOps, les étudiants acquièrent des compétences pratiques et pertinentes pour le marché du travail, leur permettant de gérer efficacement des environnements de développement et de production modernes.

**5. Préparer un environnement de test**

**Explication**: Préparer un environnement de test consiste à créer et configurer un espace où les applications peuvent être testées avant leur déploiement en production. Cela inclut la configuration de serveurs, de bases de données, et d'autres services nécessaires pour simuler l'environnement de production.

**Exemple**: Utilisation d'Ansible pour préparer un environnement de test.

yaml

Copier le code

---

- name: Préparer l'environnement de test

hosts: test\_servers

become: yes

tasks:

- name: Installer les dépendances

apt:

name: "{{ item }}"

state: present

with\_items:

- git

- docker.io

- docker-compose

- name: Cloner le dépôt de code

git:

repo: 'https://github.com/mon\_projet/test\_app.git'

dest: /var/www/test\_app

- name: Démarrer les conteneurs Docker

docker\_compose:

project\_src: /var/www/test\_app

state: present

**6. Gérer le stockage des données**

**Explication**: Gérer le stockage des données implique la mise en place et la maintenance des systèmes de stockage, tels que les bases de données, les systèmes de fichiers distribués, et les solutions de sauvegarde. Cela inclut également la gestion des accès et la sécurité des données.

**Exemple**: Utilisation de Terraform pour configurer une base de données RDS sur AWS.

hcl

Copier le code

provider "aws" {

region = "us-west-2"

}

resource "aws\_db\_instance" "mydb" {

allocated\_storage = 20

engine = "mysql"

engine\_version = "5.7"

instance\_class = "db.t2.micro"

name = "mydatabase"

username = "admin"

password = "mypassword"

parameter\_group\_name = "default.mysql5.7"

}

**7. Gérer des conteneurs**

**Explication**: Gérer des conteneurs implique la création, le déploiement, la mise à jour et la suppression des conteneurs. Cela inclut également la gestion des orchestrateurs de conteneurs comme Kubernetes pour automatiser le déploiement, la mise à l'échelle et les opérations des conteneurs d'applications.

**Exemple**: Utilisation de Kubernetes pour déployer une application conteneurisée.

yaml

Copier le code

apiVersion: v1

kind: Pod

metadata:

name: mypod

labels:

app: myapp

spec:

containers:

- name: myapp-container

image: myapp:latest

ports:

- containerPort: 80

**2. Déployer en continu une application**

**Explication**: Déployer en continu consiste à automatiser le processus de déploiement de code dans un environnement de production ou de test après chaque modification validée dans le dépôt de code. Cela permet des déploiements rapides et fréquents.

**Exemple**: Utilisation de Jenkins pour configurer un pipeline de déploiement continu.

groovy

Copier le code

pipeline {

agent any

stages {

stage('Build') {

steps {

sh 'make build'

}

}

stage('Test') {

steps {

sh 'make test'

}

}

stage('Deploy') {

steps {

sh 'make deploy'

}

}

}

}

**8. Automatiser la mise en production d’une application avec une plateforme**

**Explication**: Automatiser la mise en production consiste à utiliser des outils et des scripts pour automatiser le déploiement des applications en production, en minimisant les erreurs humaines et en augmentant l'efficacité. Les plateformes CI/CD comme Jenkins, GitLab CI, ou CircleCI sont souvent utilisées pour cela.

**Exemple**: Utilisation de GitLab CI/CD pour automatiser la mise en production.

yaml

Copier le code

stages:

- build

- test

- deploy

build\_job:

stage: build

script:

- echo "Building the application..."

- ./build\_script.sh

test\_job:

stage: test

script:

- echo "Running tests..."

- ./test\_script.sh

deploy\_job:

stage: deploy

script:

- echo "Deploying to production..."

- ./deploy\_script.sh

only:

- master

**Explications et Utilisations en Formation DevOps**

1. **Préparer un environnement de test**:
   * **Formation**: Les étudiants apprennent à configurer des environnements de test identiques à l'environnement de production pour garantir que les tests sont pertinents et fiables. Ils utilisent des outils comme Ansible ou Vagrant pour automatiser cette configuration.
2. **Gérer le stockage des données**:
   * **Formation**: Les étudiants apprennent à configurer et à gérer différentes solutions de stockage de données, en mettant l'accent sur la sécurité et la sauvegarde. Ils utilisent des outils comme Terraform pour gérer les bases de données dans le cloud.
3. **Gérer des conteneurs**:
   * **Formation**: Les étudiants apprennent à utiliser Docker pour créer et gérer des conteneurs, ainsi que Kubernetes pour orchestrer des applications conteneurisées. Ils acquièrent des compétences pratiques en déployant des applications sur des clusters Kubernetes.
4. **Déployer en continu une application**:
   * **Formation**: Les étudiants apprennent à configurer des pipelines CI/CD pour automatiser le déploiement continu des applications. Ils utilisent des outils comme Jenkins, GitLab CI ou CircleCI pour créer des pipelines de déploiement robustes.
5. **Automatiser la mise en production d’une application avec une plateforme**:
   * **Formation**: Les étudiants apprennent à utiliser des plateformes CI/CD pour automatiser la mise en production des applications. Ils configurent des pipelines pour déployer des applications en production de manière automatique et sécurisée.

En intégrant ces compétences dans une formation DevOps, les étudiants sont préparés à gérer des environnements complexes et à automatiser des processus critiques, ce qui est essentiel pour une gestion efficace et moderne des infrastructures et des applications.

**Définir et mettre en place des statistiques de services**

**Explication**: Définir et mettre en place des statistiques de services consiste à collecter, analyser et présenter des données de performance et de disponibilité des services déployés. Cela permet de surveiller l'état des services et d'identifier les éventuels problèmes avant qu'ils n'affectent les utilisateurs.

**Exemple**: Utilisation de Prometheus et Grafana pour collecter et afficher des statistiques de services.

1. **Installer Prometheus**:
   * Télécharger et exécuter Prometheus.

bash

Copier le code

# Télécharger Prometheus

wget https://github.com/prometheus/prometheus/releases/download/v2.26.0/prometheus-2.26.0.linux-amd64.tar.gz

tar xvf prometheus-2.26.0.linux-amd64.tar.gz

cd prometheus-2.26.0.linux-amd64

# Exécuter Prometheus

./prometheus --config.file=prometheus.yml

1. **Configurer un job de scraping dans prometheus.yml**:

yaml

Copier le code

global:

scrape\_interval: 15s

scrape\_configs:

- job\_name: 'node\_exporter'

static\_configs:

- targets: ['localhost:9100']

1. **Installer et configurer Grafana**:
   * Télécharger et installer Grafana.
   * Ajouter Prometheus comme source de données et créer des tableaux de bord pour visualiser les métriques collectées.

**Exploiter une solution de supervision**

**Explication**: Exploiter une solution de supervision consiste à utiliser des outils pour surveiller l'état, les performances et la disponibilité des infrastructures et des services. Cela implique de configurer des alertes et des notifications pour informer les équipes en cas de problèmes.

**Exemple**: Utilisation de Zabbix pour superviser une infrastructure.

1. **Installer Zabbix Server**:
   * Suivre les instructions d'installation de Zabbix pour le système d'exploitation choisi.
2. **Ajouter des hôtes à surveiller**:
   * Connecter des serveurs à Zabbix en installant l'agent Zabbix sur chaque serveur et en configurant Zabbix Server pour surveiller ces agents.

bash

Copier le code

# Installer l'agent Zabbix sur un serveur Linux

sudo apt-get install zabbix-agent

sudo systemctl start zabbix-agent

sudo systemctl enable zabbix-agent

1. **Configurer des éléments de surveillance**:
   * Créer des éléments (items) dans Zabbix pour collecter des métriques spécifiques comme l'utilisation CPU, la mémoire, l'espace disque, etc.
2. **Configurer des alertes et des notifications**:
   * Définir des seuils et des actions à prendre lorsqu'un seuil est dépassé, comme envoyer un email ou un message Slack à l'équipe de support.

**Superviser les services déployés**

**Explication**: Superviser les services déployés consiste à surveiller en temps réel l'état des services pour s'assurer qu'ils fonctionnent correctement. Cela inclut la surveillance des logs, des performances, et des incidents de sécurité, ainsi que la mise en place de solutions pour prévenir et résoudre les problèmes.

**Exemple**: Utilisation d'Elastic Stack (ELK) pour superviser les services déployés.

1. **Installer Elasticsearch, Logstash, et Kibana**:
   * Suivre les instructions d'installation pour chaque composant de la stack ELK.
2. **Configurer Logstash pour collecter les logs**:
   * Créer un fichier de configuration Logstash pour ingérer des logs de différentes sources.

bash

Copier le code

input {

file {

path => "/var/log/nginx/access.log"

start\_position => "beginning"

}

}

filter {

grok {

match => { "message" => "%{COMMONAPACHELOG}" }

}

}

output {

elasticsearch {

hosts => ["localhost:9200"]

index => "nginx-logs-%{+YYYY.MM.dd}"

}

}

1. **Utiliser Kibana pour visualiser et analyser les logs**:
   * Configurer des tableaux de bord dans Kibana pour afficher les logs et les métriques de performance.
   * Définir des alertes pour des anomalies ou des erreurs spécifiques.

**Utilisation de ces compétences en tant qu'Admin Système DevOps**

1. **Définir et mettre en place des statistiques de services**:
   * **Formation**: Les étudiants apprennent à utiliser des outils comme Prometheus et Grafana pour collecter et visualiser des métriques de performance. Ils configurent des job de scraping pour récupérer des métriques de différents services et créent des tableaux de bord pour visualiser les données en temps réel.
2. **Exploiter une solution de supervision**:
   * **Formation**: Les étudiants apprennent à installer et configurer des outils de supervision comme Zabbix. Ils ajoutent des hôtes à surveiller, configurent des éléments de surveillance pour collecter des métriques, et définissent des alertes pour notifier l'équipe en cas de problèmes.
3. **Superviser les services déployés**:
   * **Formation**: Les étudiants apprennent à utiliser des solutions de supervision comme Elastic Stack (ELK) pour collecter et analyser des logs. Ils configurent Logstash pour ingérer des logs, utilisent Kibana pour visualiser les données, et définissent des alertes pour détecter des anomalies ou des erreurs.

En intégrant ces compétences dans une formation DevOps, les étudiants sont préparés à surveiller efficacement les infrastructures et les services, à collecter et analyser des métriques de performance, et à réagir rapidement aux problèmes, ce qui est essentiel pour assurer la fiabilité et la performance des systèmes en production.

4o

<div class="gallery-item">

<img src="https://ik.imagekit.io/ikmedia/backlit.jpg" alt="Photo 2">

<div class="item-content">

<div class="item-text">Discover Jenkins</div>

<div class="jenkins-button" onclick="showInfo()">Jenkins</div>

<div id="jenkins-info" style="display: none;">

<p>To create a Docker image:</p>

<pre><code># Example Dockerfile

FROM base\_image

# Add your instructions here

CMD ["/path/to/your/startup/script"]</code></pre>

</div>

<script>

function showInfo() {

var info = document.getElementById("jenkins-info");

if (info.style.display === "none") {

info.style.display = "block";

} else {

info.style.display = "none";

}

}

</script>

</div>

</div>

CSS

.jenkins-button {

margin-top: 13px;

margin-left: 5px;

background-color:#ffff66 ;

padding: 10px 20px;

border: 2px solid #333;

border-radius: 5px;

background-color: #fff;

cursor: default;

transition: all 0.3s ease;

display: inline-block;

}

.jenkins-button:hover {

background-color: #4CAF50;

cursor: pointer;

}

<div class="gallery-item">

<img src="https://ik.imagekit.io/ikmedia/backlit.jpg" alt="Photo 4">

<div class="item-content">

<div class="item-text">Discover Docker</div>

<button class="button" onclick="DockerInfo()">Docker Info</button>

<div id="dockerInfo" class="docker-info">

<p>Docker is a platform for developers and sysadmins to build, share, and run applications with containers.

The use of containers to deploy applications is called containerization.

Containers are lightweight and contain everything needed to run the application,

including libraries, system tools, code, and runtime. Docker simplifies the process of building,

shipping, and running applications by providing a common framework for packaging and running them.

</p>

</div>

<script>

function DockerInfo() {

var dockerInfoDiv = document.getElementById("dockerInfo");

if (dockerInfoDiv.style.display === "none") {

dockerInfoDiv.style.display = "block";

} else {

dockerInfoDiv.style.display = "none";

}

}

</script>

</div>

</div>

<div class="gallery-item">

<img src="https://ik.imagekit.io/ikmedia/backlit.jpg" alt="Photo 2">

<div class="item-content">

<div class="item-text">Discover Jenkins</div>

<button class="button" onclick="JenkinsInfo()">Jenkins Info</button>

<div id="jenkinsInfo" class="jenkins-info">

<p>

Docker is a platform for developers and sysadmins to build, share, and run applications with containers.

The use of containers to deploy applications is called containerization.

Containers are lightweight and contain everything needed to run the application,

including libraries, system tools, code, and runtime. Docker simplifies the process of building,

shipping, and running applications by providing a common framework for packaging and running them.

</p>

</div>

<script>

function JenkinsInfo() {

var jenkinsInfoDiv = document.getElementById("jenkinsInfo");

if (jenkinsInfoDiv.style.display === "none") {

jenkinsInfoDiv.style.display = "block";

} else {

jenkinsInfoDiv.style.display = "none";

}

}

</script>

</div>

</div>

<div class="gallery-item">

<img src="https://ik.imagekit.io/ikmedia/backlit.jpg" alt="Photo 4">

<div class="item-content">

<div class="item-text">Discover Terraform</div>

<button class="button" onclick="TerraformInfo()">Terraform Info</button>

<div id="terraformInfo" class="terraform-info">

<p>

Docker is a platform for developers and sysadmins to build, share, and run applications with containers.

The use of containers to deploy applications is called containerization.

Containers are lightweight and contain everything needed to run the application,

including libraries, system tools, code, and runtime. Docker simplifies the process of building,

shipping, and running applications by providing a common framework for packaging and running them.

</p>

</div>

<script>

function TerraformInfo() {

var terraformInfoDiv = document.getElementById("terraformInfo");

if (terraformInfoDiv.style.display === "none") {

terraformInfoDiv.style.display = "block";

} else {

terraformInfoDiv.style.display = "none";

}

}

</script>

</div>

</div>

<div class="gallery-item">

<img src="https://ik.imagekit.io/ikmedia/backlit.jpg" alt="Photo 3">

<div class="item-content">

<div class="item-text">Discover Ansible</div>

<button class="button" onclick="AnsibleInfo()">Ansible Info</button>

<div id="ansibleInfo" class="ansible-info">

<p>

Docker is a platform for developers and sysadmins to build, share, and run applications with containers.

The use of containers to deploy applications is called containerization.

Containers are lightweight and contain everything needed to run the application,

including libraries, system tools, code, and runtime. Docker simplifies the process of building,

shipping, and running applications by providing a common framework for packaging and running them.

</p>

</div>

<script>

function AnsibleInfo() {

var ansibleInfoDiv = document.getElementById("ansibleInfo");

if (ansibleInfoDiv.style.display === "none") {

ansibleInfoDiv.style.display = "block";

} else {

ansibleInfoDiv.style.display = "none";

}

}

</script>

</div>

</div>

<div class="column">

<div class="content">

<div class="label">Kubernetes1</div>

<img src="https://www.docker.com/sites/default/files/d8/2019-07/Moby-logo.png" alt="Docker Logo">

<div class="product-description">Kubes8 example how to change elit.</div>

</div>

<button class="button" onclick="DockerInfo()">Docker Info</button>

<div id="dockerInfo" class="docker-info">

<p>

Docker is a platform for developers and sysadmins to build, share, and run applications with containers.

The use of containers to deploy applications is called containerization.

Containers are lightweight and contain everything needed to run the application,

including libraries, system tools, code, and runtime. Docker simplifies the process of building,

shipping, and running applications by providing a common framework for packaging and running them.

</p>

</div>

<script>

function DockerInfo() {

var dockerInfoDiv = document.getElementById("dockerInfo");

if (dockerInfoDiv.style.display === "none") {

dockerInfoDiv.style.display = "block";

} else {

dockerInfoDiv.style.display = "none";

}

}

</script>

</div>

<div class="column">

<div class="content">

<div class="label">Kubernetes 2</div>

<img src="https://www.docker.com/sites/default/files/d8/2019-07/Moby-logo.png" alt="Docker Logo">

<div class="product-description">Kubes8 example how to change elit.</div>

</div>

<button class="button" onclick="JenkinsInfo()">Jenkins Info</button>

<div id="jenkinsInfo" class="jenkins-info">

<p>

Docker is a platform for developers and sysadmins to build, share, and run applications with containers.

The use of containers to deploy applications is called containerization.

Containers are lightweight and contain everything needed to run the application,

including libraries, system tools, code, and runtime. Docker simplifies the process of building,

shipping, and running applications by providing a common framework for packaging and running them.

</p>

</div>

<script>

function JenkinsInfo() {

var jenkinsInfoDiv = document.getElementById("jenkinsInfo");

if (jenkinsInfoDiv.style.display === "none") {

jenkinsInfoDiv.style.display = "block";

} else {

jenkinsInfoDiv.style.display = "none";

}

}

</script>

</div>

<div class="column">

<div class="content">

<div class="label">Kubernetes 3</div>

<img src="https://www.docker.com/sites/default/files/d8/2019-07/Moby-logo.png" alt="Docker Logo">

<div class="product-description">Kubes8 example how to change elit.</div>

</div>

<button class="button" onclick="AnsibleInfo()">Ansible Info</button>

<div id="ansibleInfo" class="ansible-info">

<p>

Docker is a platform for developers and sysadmins to build, share, and run applications with containers.

The use of containers to deploy applications is called containerization.

Containers are lightweight and contain everything needed to run the application,

including libraries, system tools, code, and runtime. Docker simplifies the process of building,

shipping, and running applications by providing a common framework for packaging and running them.

</p>

</div>

<script>

function AnsibleInfo() {

var ansibleInfoDiv = document.getElementById("ansibleInfo");

if (ansibleInfoDiv.style.display === "none") {

ansibleInfoDiv.style.display = "block";

} else {

ansibleInfoDiv.style.display = "none";

}

}

</script>

</div>

</div>

</div>

Style -\*

ul {

list-style-type: none;

padding: 0;

margin: 0;

padding-left: 30px;

padding-right: 15px;

}

li {

display: inline-block;

margin-right: 35px;

margin-left: 15px;

padding-left: 1px; /\* Add padding to the left \*/

padding-right: 1px; /\* Add padding to the right \*/

cursor: pointer;

}

/\* Style the sub-menu \*/

ul ul {

display: none;

position: absolute;

background-color: #f9f9f9;

min-width: 25px;

box-shadow: 0px 5px 13px 0px rgba(0,0,0,0.1);

padding: 4px 0px;

}

ul li:hover > ul {

display: block;

}

/\* Style the links \*/

a {

color: black;

text-decoration: none;

}

/\* Style the links on hover \*/

a:hover {

background-color: #f0f0f0;

}

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: Arial, sans-serif;

background-color: #ccffff;

background-image: url('path/to/your/background-image.jpg');

background-size: cover; /\* Ensures the image covers the entire body \*/

background-position: center; /\* Centers the background image \*/

background-repeat: no-repeat; /\* Prevents the image from repeating \*/

}

header {

background-color: #007399;

color: white;

padding: 15px;

text-align: center;

display: flex;

align-items: center;

}

header img {

max-width: 100px;

margin-right: 10px;

}

nav {

background-color: #2980b9;

overflow: hidden;

}

nav a {

float: left;

display: block;

color: white;

text-align: center;

padding: 14px 16px;

text-decoration: none;

}

nav a:hover {

background-color: #3498db;

color: black;

}

.search-container {

margin-top: 20px;

}

.search-bar {

padding: 10px;

width: 300px;

border: none;

border-radius: 5px;

font-size: 16px;

}

.search-container {

text-align: center;

margin: 20px;

float: right;

}

.search-bar {

padding: 10px;

width: 175px;

border: none;

border-radius: 5px;

}

.search-button {

background-color: #4CAF50;

color: white;

padding: 10px 20px;

font-size: 16px;

border: none;

border-radius: 5px;

cursor: pointer;

transition: background-color 0.3s;

margin-top: 10px;

}

.search-button:hover {

background-color: #45a049;

}

.button {

display: inline-block;

padding: 10px 20px;

font-size: 16px;

text-align: center;

text-decoration: none;

border-radius: 5px;

margin: 10px;

cursor: pointer;

background-color: #2ecc71;

color: #fff;

border: none;

}

.contact-container {

margin: 20px;

padding: 20px;

background-color: white;

}

.container {

max-width: 1200px;

margin: 20px auto;

display: flex;

}

.column {

flex: 1;

box-sizing: border-box;

background-color: #ffffff;

border-radius: 8px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

margin: 10px;

}

.content {

padding: 20px;

text-align: center;

}

img {

max-width: 100%;

height: auto;

border-radius: 8px;

margin-bottom: 10px;

}

.label {

font-size: 18px;

font-weight: bold;

margin-bottom: 10px;

color: #333333;

}

.button {

background-color: #4CAF50;

color: white;

padding: 10px 20px;

font-size: 16px;

border: none;

border-radius: 5px;

cursor: pointer;

transition: background-color 0.3s;

}

.button:hover {

background-color: #45a049;

}

.banner {

position: relative;

width: 100%;

height: 150px; /\* Adjust the height as needed \*/

background-image: url('https://ik.imagekit.io/ikmedia/backlit.jpg'); /\* Replace with your image URL \*/

background-size: cover;

background-position: center;

color: #0000b3;

text-align: center;

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

}

.banner-text {

font-size: 24px;

font-weight: bold;

margin-bottom: 10px;

}

.sub-text {

font-size: 16px;

}

.gallery {

max-width: 1200px;

margin: 20px auto;

display: flex;

flex-wrap: wrap;

justify-content: space-around;

}

.gallery-item {

width: 400px;

margin: 20px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

overflow: hidden;

border-radius: 8px;

}

img {

max-width: 100%;

height: auto;

border-radius: 8px 8px 0 0;

}

.item-content {

padding: 15px;

text-align: center;

}

.item-text {

font-size: 16px;

margin-bottom: 10px;

color: #333333;

}

.button {

background-color: #4CAF50;

color: white;

padding: 10px 20px;

font-size: 16px;

border: none;

border-radius: 5px;

cursor: pointer;

transition: background-color 0.3s;

}

.button:hover {

background-color: #45a049;

}

.product {

margin-bottom: 20px;

border-bottom: 1px solid #dddddd;

padding-bottom: 15px;

}

.product-title {

font-size: 18px;

font-weight: bold;

color: #333333;

}

.product-description {

font-size: 14px;

color: #666666;

margin-bottom: 10px;

}

.button {

background-color: #4CAF50;

color: white;

padding: 10px 20px;

font-size: 16px;

border: none;

border-radius: 5px;

cursor: pointer;

transition: background-color 0.3s;

}

.button:hover {

background-color: #45a049;

}

/\*body box-sizing: border-box; \*/

/\*main {

padding: 20px;

}\*/

/\*article \*/

.article {

background-image: url(img src='https://ik.imagekit.io/ikmedia/backlit.jpg');

/\* Replace with your image URL \*/

background-size: cover;

background-position: center;

color: blue;

text-align: center;

padding: 40px;

/\*height: 100vh;\*/

display: flex;

flex-direction: column;

justify-content: center;

align-items: center;

width: 100%;

max-width: 100%;

height: auto;

}

.article p {

font-size: 18px;

margin-bottom: 20px;

}

/\* Optional: Add an overlay to improve readability of text \*/

/\*.overlay {

position: absolute;

top: 0;

left: 0;

width: 90%;

height: 70%;

background: rgba(0, 0, 0, 0.5);

}\*/

.social-buttons {

text-align: center;

margin-top: 20px;

}

.social-buttons a {

display: inline-block;

margin: 0 10px;

text-decoration: none;

}

.social-button {

background-color: #4CAF50;

color: white;

padding: 10px 20px;

font-size: 16px;

border: none;

border-radius: 5px;

cursor: pointer;

transition: background-color 0.3s;

}

.social-button:hover {

background-color: #45a049;

}

/\*footer/\*/

.container {

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

}

.column {

text-align: center;

width: 300px;

padding: 20px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

.login-button, .password-button {

background-color: #4CAF50;

color: white;

padding: 10px 20px;

font-size: 16px;

border: none;

border-radius: 5px;

cursor: pointer;

transition: background-color 0.3s;

margin-bottom: 20px;

}

.login-button:hover, .password-button:hover {

background-color: #45a049;

}

.contact-info {

font-size: 16px;

}

/\*<-->\*/

.docker-button {

margin-top: 13px;

margin-left: 10px;

padding: 10px 20px;

border: 2px solid #333;

border-radius: 5px;

background-color: #fff;

cursor: default;

transition: all 0.3s ease;

display: inline-block;

}

.docker-button:hover {

background-color: #4CAF50;;

cursor: pointer;

}

/\*<-->\*/

.jenkins-button {

margin-top: 13px;

margin-left: 5px;

background-color:#ffff66 ;

padding: 10px 20px;

border: 2px solid #333;

border-radius: 5px;

background-color: #fff;

cursor: default;

transition: all 0.3s ease;

display: inline-block;

}

.jenkins-button:hover {

background-color: #4CAF50;

cursor: pointer;

}

/\*<-->\*/

.terraform-button {

margin-top: 13px;

margin-left: 5px;

padding: 10px 20px;

border: 2px solid #333;

border-radius: 5px;

background-color: #fff;

cursor: default;

transition: all 0.4s ease;

display: inline-block;

}

.terraform-button:hover {

background-color: #ffff66;

cursor: pointer;

}

}

.devops-container {

margin: 0;

padding: 0;

height: 100%;

width: 100%;

display: flex;

justify-content: center;

align-items: center;

border: 2px solid #007399;

}

.devops-w-button {

width: 100%;

max-width: 100%; /\* Adjust this value as needed \*/

padding: 20px;

font-size: 20px;

background-color: #007399; /\* Green background \*/

color: #003366; /\* White text \*/

border: 5px solid #007399;

cursor: pointer;

text-align: center;

transition: background-color 0.3s ease;

}

.devops-w-button:hover {

background-color: #007399;

border-color: #0088cc;

/\* Darker green on hover

\*/

}

\*ù !\*\*\*\*

Terraform clickable :

<div class="gallery-item">

<img src="https://ik.imagekit.io/ikmedia/backlit.jpg" alt="Photo 4">

<div class="item-content">

<div class="item-text">Discover Terraform</div>

<div class="docker-button" onclick="showInfo()">Git</div>

<div id="docker-info" style="display: none;">

<p>To create a Docker image:</p>

<pre><code># Example Dockerfile

FROM base\_image

# Add your instructions here

CMD ["/path/to/your/startup/script"]</code></pre>

</div>

<script>

function showInfo() {

var info = document.getElementById("docker-info");

if (info.style.display === "none") {

info.style.display = "block";

} else {

info.style.display = "none";

}

}

</script>

</div>

</div>