Explore More

Subcription : Premium CDAC NOTES & MATERIAL @99

Contact to Join
Premium Group

Click to Join
Telegram Group

For More E-Notes

Join Our Community to stay Updated

TAP ON THE ICONS TO JOIN!

	codewitharrays.in freelance project available to buy contact on 8007592194	
SR.NO	Project NAME	Technology
1	Online E-Learning Platform Hub	React+Springboot+MySql
2	PG Mates / RoomSharing / Flat Mates	React+Springboot+MySql
3	Tour and Travel management System	React+Springboot+MySql
4	Election commition of India (online Voting System)	React+Springboot+MySql
5	HomeRental Booking System	React+Springboot+MySql
6	Event Management System	React+Springboot+MySql
7	Hotel Management System	React+Springboot+MySql
8	Agriculture web Project	React+Springboot+MySql
9	AirLine Reservation System / Flight booking System	React+Springboot+MySql
10	E-commerce web Project	React+Springboot+MySql
11	Hospital Management System	React+Springboot+MySql
12	E-RTO Driving licence portal	React+Springboot+MySql
13	Transpotation Services portal	React+Springboot+MySql
14	Courier Services Portal / Courier Management System	React+Springboot+MySql
15	Online Food Delivery Portal	React+Springboot+MySql
16	Muncipal Corporation Management	React+Springboot+MySql
17	Gym Management System	React+Springboot+MySql
18	Bike/Car ental System Portal	React+Springboot+MySql
19	CharityDonation web project	React+Springboot+MySql
20	Movie Booking System	React+Springboot+MySql

	freelance_Project available to buy contact on 8007592194	
21	Job Portal web project	React+Springboot+MySql
22	LIC Insurance Portal	React+Springboot+MySql
23	Employee Management System	React+Springboot+MySql
24	Payroll Management System	React+Springboot+MySql
25	RealEstate Property Project	React+Springboot+MySql
26	Marriage Hall Booking Project	React+Springboot+MySql
27	Online Student Management portal	React+Springboot+MySql
28	Resturant management System	React+Springboot+MySql
29	Solar Management Project	React+Springboot+MySql
30	OneStepService LinkLabourContractor	React+Springboot+MySql
31	Vehical Service Center Portal	React+Springboot+MySql
32	E-wallet Banking Project	React+Springboot+MySql
33	Blogg Application Project	React+Springboot+MySql
34	Car Parking booking Project	React+Springboot+MySql
35	OLA Cab Booking Portal	React+NextJs+Springboot+MySql
36	Society management Portal	React+Springboot+MySql
37	E-College Portal	React+Springboot+MySql
38	FoodWaste Management Donate System	React+Springboot+MySql
39	Sports Ground Booking	React+Springboot+MySql
40	BloodBank mangement System	React+Springboot+MySql

41	Bus Tickit Booking Project	React+Springboot+MySql
42	Fruite Delivery Project	React+Springboot+MySql
43	Woodworks Bed Shop	React+Springboot+MySql
44	Online Dairy Product sell Project	React+Springboot+MySql
45	Online E-Pharma medicine sell Project	React+Springboot+MySql
46	FarmerMarketplace Web Project	React+Springboot+MySql
47	Online Cloth Store Project	React+Springboot+MySql
48	Train Ticket Booking Project	React+Springboot+MySql
49	Quizz Application Project	JSP+Springboot+MySql
50	Hotel Room Booking Project	React+Springboot+MySql
51	Online Crime Reporting Portal Project	React+Springboot+MySql
52	Online Child Adoption Portal Project	React+Springboot+MySql
53	online Pizza Delivery System Project	React+Springboot+MySql
54	Online Social Complaint Portal Project	React+Springboot+MySql
55	Electric Vehical management system Project	React+Springboot+MySql
56	Online mess / Tiffin management System Project	React+Springboot+MySql
57		React+Springboot+MySql
58		React+Springboot+MySql
59		React+Springboot+MySql
60		React+Springboot+MySql

Spring Boot + React JS + MySQL Project List

Sr.No	Project Name	YouTube Link
1	Online E-Learning Hub Platform Project	https://youtu.be/KMjyBaWmgzg?si=YckHuNzs7eC84-IW
2	PG Mate / Room sharing/Flat sharing	https://youtu.be/4P9clHg3wvk?si=4uEsi0962CG6Xodp
3	Tour and Travel System Project Version 1.0	https://youtu.be/-UHOBywHaP8?si=KHHfE_A0uv725f12
4	Marriage Hall Booking	https://youtu.be/VXz0kZQi5to?si=llOS-QG3TpAFP5k7
5	Ecommerce Shopping project	https://youtu.be/vJ_C6LkhrZ0?si=YhcBylSErvdn7paq
6	Bike Rental System Project	https://youtu.be/FIzsAmIBCbk?si=7ujQTJqEgkQ8ju2H
7	Multi-Restaurant management system	https://youtu.be/pvV-pM2Jf3s?si=PgvnT-yFc8ktrDxB
8	Hospital management system Project	https://youtu.be/IynIouBZvY4?si=CXzQs3BsRkjKhZCw
9	Municipal Corporation system Project	https://youtu.be/cVMx9NVyI4I?si=qX0oQt-GT-LR_5jF
10	Tour and Travel System Project version 2.0	https://youtu.be/_4u0mB9mHXE?si=gDiAhKBowi2gNUKZ

Sr.No	Project Name	YouTube Link
11	Tour and Travel System Project version 3.0	https://youtu.be/Dm7nOdpasWg?si=P_Lh2gcOFhlyudug
12	Gym Management system Project	https://youtu.be/J8_7Zrkg7ag?si=LcxV51ynfUB7OptX
13	Online Driving License system Project	https://youtu.be/3yRzsMs8TLE?si=JRI_z4FDx4Gmt7fn
14	Online Flight Booking system Project	https://youtu.be/m755rOwdk8U?si=HURvAY2VnizIyJlh
15	Employee management system project	https://youtu.be/lD1iE3W_GRw?si=Y_jv1xV_BljhrD0H
16	Online student school or college portal	https://youtu.be/4A25aEKfei0?si=RoVgZtxMk9TPdQvD
17	Online movie booking system project	https://youtu.be/Lfjv_U74SC4?si=fiDvrhhrjb4KSlSm
18	Online Pizza Delivery system project	https://youtu.be/Tp3izreZ458?si=8eWAOzA8SVdNwlyM
19	Online Crime Reporting system Project	https://youtu.be/0UlzReSk9tQ?si=6vN0e70TVY1GOwPO
20	Online Children Adoption Project	https://youtu.be/3T5HC2HKyT4?si=bntP78niYH802I7N

Q - 1) What is Kubernetes, and what are its main features?

- Kubernetes is an open-source container orchestration platform used for automating the deployment, scaling, and management of containerized applications.
- Main features of Kubernetes:
  - Automated Scheduling: Efficiently schedules containers across a cluster based on resource requirements and constraints.
  - Self-Healing: Automatically restarts failed containers, replaces containers, and reschedules when nodes fail.
  - Horizontal Scaling: Scales applications up and down based on demand.
  - Load Balancing and Service Discovery: Distributes network traffic to ensure the application is stable.
  - Automated Rollbacks and Rollouts: Manages software updates and rollbacks efficiently.

Q - 2) What is a Kubernetes cluster, and what are its main components?

- A Kubernetes cluster is a set of nodes used to run containerized applications. It consists of a control plane and worker nodes.
- Main components:
  - Control Plane: Manages the Kubernetes cluster. Includes components like the API server, etcd, controller manager, and scheduler.
  - Worker Nodes: Execute workloads and run containerized applications. Includes components like kubelet, kube-proxy, and container runtime.

Q - 3) Explain the role of the Kubernetes master node and its components.

- The master node is responsible for managing the Kubernetes cluster and controlling the scheduling of Pods.
- Main components of the master node:
  - API Server: Serves as the frontend for the Kubernetes control plane.
  - etcd: A key-value store that stores the configuration data and state of the cluster.
  - Controller Manager: Runs various controllers that manage the cluster's state, like node controllers and endpoint controllers.
  - Scheduler: Determines which nodes the Pods should run on based on resource availability and constraints.

Q - 4) What are Pods in Kubernetes, and how do they differ from containers?

- A Pod is the smallest and most basic deployable unit in Kubernetes. It can contain one or more tightly coupled containers that share the same network and storage.
- Difference from containers:
  - Containers within a Pod share the same IP address and port space, enabling them to communicate with each other using `localhost`.
  - Pods provide a higher level of abstraction than containers, allowing multiple containers to work together as a single entity.

Q - 5) How do you create and manage Pods in Kubernetes?

- Creating a Pod: You can create a Pod using the `kubectl` command or by defining a Pod manifest file (YAML or JSON).

```
kubectl run my-pod --image=my-image
```

- Managing Pods: Use commands like `kubectl get pods`, `kubectl describe pod <pod-name>`, and `kubectl delete pod <pod-name>` to manage Pods.

Q - 6) What is a ReplicaSet in Kubernetes, and why is it used?

- A ReplicaSet ensures that a specified number of replica Pods are running at any given time.
- Purpose: It maintains the desired state of application replicas, replacing any failed Pods to keep the cluster in a stable state.

Q - 7) How does a Deployment work in Kubernetes, and how is it different from a ReplicaSet?

- A Deployment manages ReplicaSets and provides declarative updates to applications.
- Difference:
  - While ReplicaSets only ensure that a specified number of replicas are running, Deployments offer advanced features like rolling updates, rollbacks, and scaling.

Q - 8) What are Services in Kubernetes, and what are the different types of Services available?

- Services in Kubernetes provide a stable endpoint to access Pods, abstracting away the dynamic nature of Pod IP addresses.
- Types of Services:
  - ClusterIP: Exposes the service to internal cluster communication.

- NodePort: Exposes the service on a static port on each node's IP.
- LoadBalancer: Uses an external load balancer to expose the service to the outside world.
- ExternalName: Maps a service to a DNS name.

Q - 9) How do you expose a Kubernetes service to external traffic?

- You can expose a service to external traffic using a LoadBalancer service type or an Ingress resource.

```
kubectl expose deployment my-deployment --type=LoadBalancer --
port=80
```

- Ingress controllers can also be used to manage external access to services.

Q - 10) What is the difference between a ClusterIP, NodePort, and LoadBalancer service?

- ClusterIP: Default service type. Exposes the service on an internal IP, making it accessible only within the cluster.
- NodePort: Exposes the service on a specific port on each node's IP, allowing external access to the service via the node IP and port.
- LoadBalancer: Creates an external load balancer that distributes incoming traffic to the Pods, ideal for production environments when exposing services to the internet.

Q - 11) What are Namespaces in Kubernetes, and how are they used?

Namespaces in Kubernetes are used to organize objects in a cluster into separate groups. They provide a way to divide cluster resources among multiple users or teams.

- Use cases:
  - To create separate environments (e.g., development, testing, production) within a single cluster.
  - To manage resources more efficiently and apply access control policies within specific namespaces.

Q - 12) How do you scale a deployment in Kubernetes?

- Scaling a deployment in Kubernetes can be done using the `kubectl` command or by modifying the deployment's configuration.
- Using `kubectl` command:

```
kubectl scale deployment my-deployment --replicas=5
```

- This command scales the deployment to 5 replicas. Kubernetes will automatically create or terminate Pods to match the desired count.

Q - 13) What is a ConfigMap, and how is it used in Kubernetes?

- A ConfigMap is used to store configuration data in key-value pairs that can be used by Pods and containers.
- Use cases:
  - To decouple configuration artifacts from container images, making applications more portable and easy to manage.
  - ConfigMaps can be injected into a Pod's environment variables or mounted as files in a container's filesystem.

Q - 14) What are Secrets in Kubernetes, and how are they different from ConfigMaps?

- Secrets in Kubernetes are used to store sensitive data, such as passwords, API keys, and tokens, securely.
- Difference from ConfigMaps:
  - Secrets are encrypted and are stored in a way that keeps the data secure.
  - ConfigMaps are used for non-sensitive configuration data, while Secrets are specifically designed to handle sensitive information.

Q - 15) .Explain the concept of StatefulSets in Kubernetes.

- A StatefulSet is a Kubernetes resource used to manage the deployment and scaling of stateful applications.
- Characteristics:
  - Each Pod in a StatefulSet has a unique, persistent identity.
  - StatefulSets maintain the order and uniqueness of Pods, ensuring they are created and deleted in a specific sequence.

Q - 16) What is the difference between a Deployment and a StatefulSet?

- Deployment:
  - Suitable for stateless applications where the identity of Pods doesn't matter.
  - Pods are identical and can be easily replaced.
- StatefulSet:
  - Used for stateful applications where Pod identity and order are crucial.

- Each Pod has a unique identifier, and the scaling operations respect the order of creation and deletion.

Q - 17) How do you perform rolling updates and rollbacks in Kubernetes?

- Rolling updates allow you to update the application in a controlled manner without downtime.

```
kubectl set image deployment/my-deployment my-container=my-image:v2
```

- Rollbacks can be performed if the new deployment causes issues:

```
kubectl rollout undo deployment/my-deployment
```

- These commands ensure seamless updates while maintaining high availability.

Q - 18) What are DaemonSets, and when would you use them?

- A DaemonSet ensures that a copy of a specific Pod runs on all (or some) nodes in the Kubernetes cluster.
- Use cases:
  - Running monitoring, logging, or networking services on every node.
  - Useful for running background services like Fluentd, Logstash, or other data collection agents.

Q - 19) Explain the use of Ingress in Kubernetes.

- Ingress in Kubernetes is used to manage external access to services within the cluster, typically HTTP/HTTPS traffic.
- It provides a way to define routing rules, SSL termination, and load balancing, making it more flexible than using NodePort or LoadBalancer services.

Q - 20) How do you secure communication between Pods in a Kubernetes cluster?

- Communication between Pods in a Kubernetes cluster can be secured using the following methods:
  - Network Policies: Define rules to control traffic between Pods, specifying which Pods can communicate with each other.
  - TLS/SSL Encryption: Use certificates to encrypt data in transit between services.
  - Service Meshes (like Istio): Provide advanced security features, such as mutual TLS (mTLS) and policy-based traffic management.

Q - 21) What is the purpose of the `kubectl` command-line tool in Kubernetes?

- `kubectl` is the command-line tool for interacting with Kubernetes clusters.
- It allows users to deploy and manage applications, inspect cluster resources, view logs, and troubleshoot issues.
- Common commands include `kubectl get`, `kubectl create`, `kubectl delete`, and `kubectl describe`.

Q - 22) How do you manage persistent storage in Kubernetes?

- Kubernetes manages persistent storage using Persistent Volumes (PVs) and Persistent Volume Claims (PVCs).
- PVs represent physical storage available to the cluster, while PVCs are requests for that storage made by applications.
- Storage can be managed using different backends like local storage, NFS, AWS EBS, Google Persistent Disk, etc.

Q - 23) What are Persistent Volumes (PVs) and Persistent Volume Claims (PVCs)?

- Persistent Volumes (PVs): Storage resources in the cluster, defined by administrators, that are independent of individual Pods.
- Persistent Volume Claims (PVCs): Requests for storage resources by a user or a Pod. They specify the size and access mode of the desired storage.
- PVCs are bound to PVs, allowing Pods to use storage that remains available even if the Pod is deleted or recreated.

Q - 24) How does Kubernetes handle resource limits and requests for CPU and memory?

- Kubernetes uses resource requests and limits to manage CPU and memory allocation for containers.
  - Resource Requests: Specify the minimum amount of resources a container needs. The scheduler uses these values to assign Pods to nodes.
  - Resource Limits: Define the maximum amount of resources a container can use. If the container exceeds these limits, it may be throttled or terminated.
- These settings help prevent resource overuse and ensure fair allocation among all containers.

Q - 25) What are Taints and Tolerations in Kubernetes?

- Taints are applied to nodes to prevent certain Pods from being scheduled on them.
- Tolerations allow Pods to be scheduled on nodes with matching taints.

- This mechanism is used to control how Pods are placed on specific nodes and to ensure that certain nodes are reserved for specific workloads.

Q - 26) How do you troubleshoot a failing Pod in Kubernetes?

- To troubleshoot a failing Pod, you can use the following steps:

    1. Check the Pod's status: Use `kubectl get pods` to see if the Pod is in a failed or pending state.
    2. Describe the Pod: Use `kubectl describe pod <pod-name>` to get detailed information about the Pod's events and reasons for failure.
    3. View logs: Use `kubectl logs <pod-name>` to view the container logs for error messages.
    4. Check container status: Use `kubectl get events` for recent events related to the Pod or node.

Q - 27) What is Helm, and how does it help in managing Kubernetes applications?

- Helm is a package manager for Kubernetes that simplifies the deployment and management of applications.
- It uses Helm charts, which are pre-configured templates that define Kubernetes resources.
- Helm helps automate the deployment, upgrade, and rollback of applications in a consistent and reusable manner.

Q - 28) What is a Kubernetes Operator, and how does it differ from Helm charts?

- A Kubernetes Operator is a software extension that uses custom resources to manage applications and their components.
- Difference from Helm charts:
    - Helm charts are mainly used for simple deployments and upgrades.
    - Operators handle more complex logic, such as application lifecycle management, self-healing, backups, and scaling, which go beyond Helm's capabilities.

Q - 29) How does Kubernetes handle container orchestration and scheduling?

- Kubernetes uses a scheduler to decide which node will run a Pod based on resource availability, affinity/anti-affinity rules, and other constraints.

- The orchestration process involves automatically managing the deployment, scaling, and operation of containers to ensure that the desired state of the application is maintained.

Q - 30) What are the main differences between Kubernetes and Docker Swarm?

- Kubernetes:
  - More complex with a steeper learning curve but provides advanced features for orchestration and scalability.
  - Supports extensive tooling, custom resource definitions, and robust service discovery.
  - Preferred for large-scale, production-grade deployments.
- Docker Swarm:
  - Easier to set up and simpler to use, but lacks some of the advanced features of Kubernetes.
  - Integrated with Docker, making it ideal for smaller-scale deployments.
  - Limited capabilities in handling complex configurations and self-healing.

https://www.youtube.com/@codewitharrays

https://www.instagram.com/codewitharrays/

https://t.me/codewitharrays Group Link: https://t.me/ccee2025notes

+91 8007592194 +91 9284926333

codewitharrays@gmail.com

https://codewitharrays.in/project