**Day-Wise Detailed Study Plan**

This plan follows a progression from **basic concepts** to more **advanced scenarios**, with practical exercises each day to build real-world knowledge.

**Day 1: Core Concepts & Kubernetes Cluster Basics**

* **Topics:**
  + What is Kubernetes?
  + Cluster Architecture: Master & Worker nodes, etcd, Control Plane components (API Server, Controller Manager, Scheduler, Kubelet, Kube Proxy).
  + Kubernetes objects: Pods, Deployments, ReplicaSets, ConfigMaps, Secrets.
* **Activities:**
  + Watch introductory videos/tutorials on Kubernetes basics (e.g., Kubernetes.io docs or KodeKloud CKA).
  + Set up Minikube/Kind or a practice lab environment.
  + Create and list Pods, Deployments, and Services using kubectl.
* **Practice:**
  + Create a Pod using YAML, exposing it via a Service.
  + Explore the kubectl get, describe, and logs commands.

**Day 2: Cluster Installation & Configuration**

* **Topics:**
  + Install Kubernetes with kubeadm (Master and Worker nodes).
  + Configure cluster networking (CNI Plugins: Flannel, Calico).
  + Secure Cluster Communication (certificates, TLS).
* **Activities:**
  + Set up a 3-node cluster using kubeadm.
  + Explore kubeconfig and modify contexts.
* **Practice:**
  + Verify node communication (kubectl get nodes).
  + Deploy a network plugin (e.g., Flannel).

**Day 3: Kubernetes API Primitives & Resource Management**

* **Topics:**
  + Pods, ReplicaSets, Deployments, Namespaces.
  + Resource limits and requests.
  + ConfigMaps and Secrets.
* **Activities:**
  + Deploy a Deployment with 3 replicas.
  + Use ConfigMaps and Secrets for environment variables and sensitive data.
* **Practice:**
  + Create a Deployment that uses ConfigMaps for environment variables.
  + Assign resource requests/limits to a Pod.

**Day 4: Scheduling**

* **Topics:**
  + Labels, Selectors, and Node Affinity.
  + Taints and Tolerations.
  + Manual scheduling (static Pods).
  + DaemonSets and multiple schedulers.
* **Activities:**
  + Apply labels and schedule Pods based on NodeSelector.
  + Deploy a DaemonSet.
* **Practice:**
  + Write a YAML manifest for a Pod that tolerates a taint.
  + Create a static Pod using kubelet.

**Day 5: Networking (Part 1)**

* **Topics:**
  + Kubernetes networking basics.
  + Cluster IP, NodePort, LoadBalancer Services.
  + Ingress controllers.
* **Activities:**
  + Expose an application using a Service.
  + Configure an Ingress resource with custom rules.
* **Practice:**
  + Deploy an NGINX Pod and expose it via Ingress.
  + Use kubectl port-forward to access the Pod locally.

**Day 6: Networking (Part 2)**

* **Topics:**
  + CoreDNS.
  + Network policies (restricting traffic).
  + Troubleshooting networking issues.
* **Activities:**
  + Apply a network policy to allow only specific Pods to communicate.
  + Test DNS resolution using busybox Pods.
* **Practice:**
  + Create a Pod with a restrictive network policy.

**Day 7: Application Lifecycle Management**

* **Topics:**
  + Rolling updates and rollbacks.
  + Scaling applications.
  + Self-healing (Liveness and Readiness Probes).
* **Activities:**
  + Deploy a web application with liveness and readiness probes.
  + Perform rolling updates and rollbacks.
* **Practice:**
  + Simulate a failing probe and observe Pod behavior.
  + Scale a Deployment up and down.

**Day 8: Storage**

* **Topics:**
  + Persistent Volumes (PVs) and Persistent Volume Claims (PVCs).
  + Dynamic provisioning with StorageClasses.
  + Access modes for Volumes.
* **Activities:**
  + Create a PV and bind it to a PVC.
  + Configure a StatefulSet with persistent storage.
* **Practice:**
  + Deploy a database (e.g., MySQL) with persistent storage.
  + Experiment with ReadWriteOnce and ReadOnlyMany modes.

**Day 9: Security**

* **Topics:**
  + Authentication & Authorization (RBAC).
  + Security contexts.
  + Network policies (advanced).
  + Image security.
* **Activities:**
  + Create a ServiceAccount and bind it to a Pod.
  + Apply RBAC policies to restrict access to namespaces.
* **Practice:**
  + Configure a Pod to run as a non-root user.
  + Deploy a Pod with a restricted security context.

**Day 10: Logging & Monitoring**

* **Topics:**
  + Application and cluster logging.
  + Monitor cluster components using metrics-server.
* **Activities:**
  + Deploy the metrics-server and visualize resource usage.
  + Configure centralized logging using Fluentd.
* **Practice:**
  + Monitor CPU/Memory usage of Pods using kubectl top.
  + Inspect logs for failed Pods.

**Day 11: Cluster Maintenance**

* **Topics:**
  + Cluster upgrades.
  + Backups and restore (etcd).
  + Node maintenance and draining.
* **Activities:**
  + Simulate a node drain and cordon operation.
  + Back up and restore etcd data.
* **Practice:**
  + Upgrade kubeadm/kubelet on a cluster.
  + Simulate a control plane component failure and restore it.

**Day 12: Troubleshooting (Part 1)**

* **Topics:**
  + Troubleshooting application failures.
  + Troubleshooting Pods stuck in Pending or CrashLoopBackOff.
* **Activities:**
  + Debug a Pod using logs and events.
  + Use kubectl debug to inspect container issues.
* **Practice:**
  + Resolve a Pod stuck due to resource constraints.

**Day 13: Troubleshooting (Part 2)**

* **Topics:**
  + Troubleshooting cluster networking.
  + Troubleshooting control plane components.
* **Activities:**
  + Simulate and resolve a network failure.
  + Restart control plane components manually.
* **Practice:**
  + Debug a failing kube-proxy Pod.

**Day 14: Mock Tests and Review**

* **Activities:**
  + Attempt mock tests on platforms like Killer.sh.
  + Review all YAML manifests created so far.
* **Practice:**
  + Solve 50-70 mock questions under timed conditions.

**Day 15: Final Revision and Practice Exam**

* **Activities:**
  + Revise critical concepts (RBAC, Networking, Troubleshooting).
  + Take a full-length practice exam.
* **Practice:**
  + Revisit weaker topics based on mock exam results.

**Additional Resources**

1. **Official Kubernetes Documentation**: https://kubernetes.io/docs/
2. **Practice Labs**: KodeKloud, Katacoda, Play with Kubernetes.
3. **Mock Exams**: Killer.sh, CKA Prep Labs.
4. **Books**: *Kubernetes Up & Running* by Kelsey Hightower.

* **Core Concepts**
* How do you create a Pod with a custom label and expose it via a ClusterIP service?
* A Deployment is not scaling beyond 3 replicas. What could be the reason? How do you fix it?
* You need to list all Pods running in a specific namespace. Write the command.
* A developer deleted a Pod by mistake. How do you ensure that Pods in a Deployment are automatically recreated if deleted?
* How do you create a ConfigMap with multiple key-value pairs from a file?
* **Cluster Architecture**
* Describe the role of etcd in a Kubernetes cluster. How would you back up its data?
* The API server is not responding to kubectl commands. What steps would you take to troubleshoot this?
* What happens if the kube-proxy service on a node is down? How do you identify and resolve the issue?
* You need to set up a High Availability Kubernetes control plane. What are the steps to achieve this?
* How do you check the status of control plane components like kube-apiserver and controller-manager?
* **Scheduling**
* How do you assign a Pod to a specific node in the cluster using labels?
* A Pod is in Pending state. How do you check why it wasn’t scheduled?
* A cluster has multiple nodes with varying resource capacities. How can you ensure that Pods with high CPU requirements are scheduled on the right node?
* Explain how Taints and Tolerations work. Write a YAML manifest to schedule a Pod on a tainted node.
* Describe the use of DaemonSets. How do you create one to run on all nodes in the cluster?
* **Networking**
* Your Pods cannot communicate with one another. How do you diagnose and resolve the issue?
* Write a YAML file to create a NetworkPolicy that only allows traffic from Pods with a specific label.
* How do you expose an application running on a Pod to the outside world using a LoadBalancer service?
* You need to set up an Ingress to route traffic to different backend services based on URL paths. Write an Ingress manifest.
* A Pod is unable to resolve DNS. What steps do you take to identify and fix the issue?
* **Storage**
* How do you create a PersistentVolume (PV) and PersistentVolumeClaim (PVC) to store data?
* A StatefulSet needs to store its data persistently. How would you configure storage for it?
* How do you verify that a PVC is correctly bound to a PV?
* Write a YAML file to create a Pod that uses a PVC for storage.
* Explain the difference between ReadWriteOnce and ReadOnlyMany access modes.
* **Application Lifecycle Management**
* How do you perform a rolling update on a Deployment without downtime?
* A Deployment rollback is required. How do you achieve this, and how do you check the revision history?
* Write a YAML file to configure liveness and readiness probes for an application.
* An application needs to scale based on CPU utilization. How do you configure autoscaling?
* Describe how Kubernetes handles self-healing for Pods in a ReplicaSet.
* **Security**
* How do you create a ServiceAccount and bind it to a specific Pod?
* Write a YAML file to configure a Pod to run as a non-root user.
* What are the steps to restrict a Pod's communication to specific IP ranges using a NetworkPolicy?
* A container image has a security vulnerability. How do you ensure Pods use only signed images from a trusted registry?
* Describe the role of RBAC in Kubernetes. How do you create a RoleBinding for a specific user?
* **Logging and Monitoring**
* How do you check the logs for a Pod that is in CrashLoopBackOff state?
* Describe how you would set up centralized logging for all cluster components.
* How do you monitor resource usage for Pods in a namespace using kubectl commands?
* The metrics-server is not returning resource usage data. What steps do you take to fix it?
* Write a script to fetch the logs of all Pods in a namespace.
* **Cluster Maintenance**
* A cluster upgrade is required. How do you upgrade kubeadm, kubelet, and kubectl on all nodes?
* What is the process for draining a node for maintenance?
* Describe how you would back up and restore the etcd database.
* How do you troubleshoot a node that has gone into NotReady state?
* Write the commands to add a new node to an existing cluster.
* **Troubleshooting**
* A Pod is stuck in ContainerCreating state. How do you troubleshoot the issue?
* A control plane component, such as the scheduler, is not running. How do you debug and resolve the problem?
* How do you identify which nodes are causing network failures in the cluster?
* An application is unable to connect to its database Pod. What steps do you take to debug the issue?
* Write a series of kubectl commands to debug a failed Deployment.
* **Installation, Configuration & Validation**
* How do you set up a Kubernetes cluster with kubeadm?
* Write a YAML file to create a custom ResourceQuota for a namespace.
* A developer wants to use a specific CNI plugin (e.g., Calico). How do you install it in a cluster?
* How do you verify that all cluster nodes have the correct Kubernetes version after an upgrade?
* What are the steps to validate that a new cluster installation is working correctly?
* **Advanced Scenarios**
* A Pod is not being terminated during a Deployment update. How do you debug it?
* Describe how you would configure a HorizontalPodAutoscaler for a Deployment.
* How do you create a StatefulSet with headless service DNS?
* An Ingress controller is not routing traffic correctly. How do you debug the issue?
* Write a manifest to configure a Pod with multiple containers, sharing the same volume.
* **Mock Questions for Real-Time Practice**
* How do you recover a control plane node from failure using etcd snapshots?
* What steps do you take to troubleshoot image pull errors in a Pod?
* How do you manually schedule a Pod to a specific node without a scheduler?
* Write a YAML file for a Pod that uses a ConfigMap and Secret as environment variables.
* How do you check the resource usage of control plane components?
* **Challenging Real-World Scenarios**
* An application has high latency. How do you debug the issue at the network and application levels?
* You need to create a custom scheduler for a specific set of Pods. Describe the process.
* Write a YAML file to configure a Pod with init containers.
* A StatefulSet is not creating Pods. What are the possible reasons?
* How do you resolve issues where kubelet cannot start on a worker node?
* **Deep-Dive into Kubernetes Topics**
* Explain the difference between DaemonSets and Deployments. Provide YAML examples for both.
* How do you debug a Pod stuck in ImagePullBackOff state?
* Write a YAML manifest to configure Ingress with TLS.
* How do you restore a cluster from a disaster where all control plane nodes are lost?
* Describe the role of kube-proxy in Service networking.
* **General Exam-Oriented Questions**
* How do you apply a custom Helm chart for an application?
* Write the steps to debug a PersistentVolumeClaim that is stuck in Pending state.
* What are the differences between Ingress and LoadBalancer services?
* How do you ensure that Pods in a specific namespace cannot access Pods in another namespace?
* Write a YAML manifest for a Pod that uses a hostPath volume.
* **Final Scenarios to Simulate**
* Set up a Pod with specific CPU and memory resource limits.
* An application Deployment is failing with insufficient CPU. How do you resolve it?
* How do you create a Kubernetes Job for a batch processing task?
* A Pod is not terminating properly. How do you debug the issue?
* Describe how you would migrate an application to another namespace.