# Chat Transcript

## User

1) How do you run an Nginx application in Docker?  
  
2) Do you have experience working with APIs?  
  
3) You have a Node.js application running on port 8081 – can you write a Dockerfile for it?  
  
4) Do you have experience with Kubernetes?  
  
5) Based on the Dockerfile, can you write a Kubernetes Deployment file for the same Node.js app?  
  
6) How would you rate yourself in Linux?  
  
7) Can you explain some commonly used Linux commands?  
  
8) What are the steps to create a VPC Peering connection in AWS?  
  
9) You have an EC2 instance in a private subnet – how will you access it?  
  
10) Write a Terraform script to create an EC2 instance and an S3 bucket.  
  
11) How can you monitor an EC2 instance application using CloudWatch?

## Assistant

Great set of questions! Let’s go through each one step by step:  
  
1) How do you run an Nginx application in Docker?  
To run Nginx in Docker:  
- Create a Dockerfile (optional) if you want to customize Nginx.  
- Use the official Nginx image for a quick setup:  
 docker run --name my-nginx -p 80:80 -d nginx  
To serve custom content, mount a volume:  
 docker run --name my-nginx -p 80:80 -v /path/to/html:/usr/share/nginx/html:ro -d nginx  
  
2) Do you have experience working with APIs?  
Yes, I can help with:  
- Designing RESTful APIs  
- Consuming APIs using tools like curl, Postman, or in code (Node.js, Python, etc.)  
- Securing APIs with tokens, OAuth  
- Monitoring and logging API usage  
  
3) Dockerfile for a Node.js application running on port 8081  
Here’s a sample Dockerfile:  
FROM node:18  
WORKDIR /app  
COPY package\*.json ./  
RUN npm install  
COPY . .  
EXPOSE 8081  
CMD ["node", "app.js"]  
  
4) Do you have experience with Kubernetes?  
Yes, I can help with:  
- Creating Deployments, Services, ConfigMaps, Secrets  
- Helm charts  
- Autoscaling and rolling updates  
- Monitoring with Prometheus/Grafana  
- Troubleshooting pods and networking  
  
5) Kubernetes Deployment file for the Node.js app  
Here’s a basic deployment.yaml:  
apiVersion: apps/v1  
kind: Deployment  
metadata:  
 name: nodejs-app  
spec:  
 replicas: 2  
 selector:  
 matchLabels:  
 app: nodejs-app  
 template:  
 metadata:  
 labels:  
 app: nodejs-app  
 spec:  
 containers:  
 - name: nodejs-app  
 image: your-dockerhub-username/nodejs-app:latest  
 ports:  
 - containerPort: 8081  
---  
apiVersion: v1  
kind: Service  
metadata:  
 name: nodejs-service  
spec:  
 selector:  
 app: nodejs-app  
 ports:  
 - protocol: TCP  
 port: 80  
 targetPort: 8081  
 type: LoadBalancer  
  
6) How would you rate yourself in Linux?  
I can assist with most Linux tasks including:  
- File system navigation  
- Process management  
- Networking  
- Shell scripting  
- System administration  
  
7) Commonly used Linux commands  
Here are a few:  
- ls – List files  
- cd – Change directory  
- pwd – Print working directory  
- cp, mv, rm – Copy, move, remove files  
- top, htop – Monitor processes  
- ps, kill – Process management  
- chmod, chown – Permissions  
- grep, awk, sed – Text processing  
- curl, wget – Network requests  
- df, du – Disk usage  
  
8) Steps to create a VPC Peering connection in AWS  
1. Go to VPC Dashboard.  
2. Click Peering Connections > Create Peering Connection.  
3. Choose requester and accepter VPCs.  
4. Click Create.  
5. Accept the peering request in the accepter VPC.  
6. Update route tables in both VPCs.  
7. Modify security groups if needed.  
  
9) Accessing an EC2 instance in a private subnet  
Options:  
- Use a Bastion Host in a public subnet.  
- Use Systems Manager (SSM) if the instance has SSM agent and IAM role.  
- VPN or Direct Connect to the VPC.  
  
10) Terraform script to create EC2 and S3 bucket  
provider "aws" {  
 region = "us-east-1"  
}  
resource "aws\_instance" "example" {  
 ami = "ami-0c55b159cbfafe1f0"  
 instance\_type = "t2.micro"  
 tags = {  
 Name = "ExampleInstance"  
 }  
}  
resource "aws\_s3\_bucket" "example\_bucket" {  
 bucket = "my-unique-bucket-name-12345"  
 acl = "private"  
}  
  
11) Monitoring EC2 application using CloudWatch  
Steps:  
1. Install CloudWatch Agent on EC2.  
2. Configure agent to monitor logs, metrics.  
3. Use cloudwatch-agent-config.json to define metrics.  
4. Start the agent.  
5. View metrics and logs in CloudWatch dashboard.  
6. Set up Alarms for thresholds.  
  
Would you like help setting up any of these in your environment?

Aws lambda is used to perform quick actions and small amount of workloads where as aws fargate is used running containers compute server (takes care of worker node ) eks (master node)

Ingress will route the traffic