CLOUD ROADMAP

120 concepts to become a cloud engineer

8 areas, 15 concepts each. Every concept explained in one line, the order to study each area, and two practice questions to test yourself.

How to use it: one area at a time, tick each concept only when you can explain it out loud in under a minute, then answer the two practice questions without notes.

@chalkesahil · chalkesahil.ai

AREA 01 / 08

Linux and networking

Study order: Every cloud server is a Linux box on a network; start here.

01	**Linux file system**	Where configs, logs and apps live.
02	**Shell commands**	Move around and edit files fast.
03	**Permissions**	Users, groups and chmod.
04	**Processes and services**	systemctl, ps and kill.
05	**SSH**	Secure login with keys.
06	**Bash scripting**	Automate repeated setup.
07	**IP addresses**	Public, private and CIDR ranges.
08	**Subnets**	Split a network into parts.
09	**DNS**	Names to IP addresses.
10	**HTTP and HTTPS**	How web traffic moves.
11	**Ports**	Which service listens where.
12	**Firewalls**	Allow or block traffic.
13	**Load balancing basics**	Spread traffic over servers.
14	**curl and dig**	Debug from the terminal.
15	**Logs**	Find what broke and when.

Practice

1. A website loads by IP but not by name. What do you check?
2. Write a script that restarts a service if it stops.

AREA 02 / 08

Cloud basics

Study order: Pick AWS first; most Indian job posts ask for it.

01	**What cloud is**	Renting servers instead of buying them.
02	**IaaS, PaaS, SaaS**	Who manages what.
03	**Regions and zones**	Where your servers physically run.
04	**Shared responsibility**	What AWS secures vs what you secure.
05	**Free tier**	Practise without a big bill.
06	**Console vs CLI**	Click vs script.
07	**AWS CLI**	Manage resources from the terminal.
08	**Pricing models**	On-demand, reserved and spot.
09	**Well-Architected Framework**	AWS's checklist for good design.
10	**High availability**	Survive a zone going down.
11	**Scalability**	Grow with traffic.
12	**Disaster recovery**	Backups and restore plans.
13	**Tagging**	Know who owns each resource.
14	**Azure and GCP**	Same ideas, different names.
15	**AWS Cloud Practitioner**	The usual first certificate.

Practice

1. Explain the shared responsibility model with one example.
2. How would you keep an app running if one zone fails?

AREA 03 / 08

Compute

Study order: Launch a server, break it, and fix it yourself.

01	**EC2**	Virtual servers on AWS.
02	**Instance types**	CPU, memory and cost trade-offs.
03	**AMIs**	Saved server images.
04	**User data**	Run setup when a server starts.
05	**Security groups**	Firewall for each server.
06	**Elastic IPs**	A fixed public IP.
07	**Auto Scaling groups**	Add servers when traffic grows.
08	**Load balancers**	ALB in front of many servers.
09	**Health checks**	Replace broken servers automatically.
10	**Lambda**	Run code without managing servers.
11	**Serverless trade-offs**	Cold starts and limits.
12	**Elastic Beanstalk**	Deploy without setting up servers.
13	**Spot instances**	Cheap servers that can vanish.
14	**Right-sizing**	Pay only for what you use.
15	**Bastion hosts**	Safe access to private servers.

Practice

1. Traffic doubles every evening. How do you handle it?
2. When would you choose Lambda over EC2?

AREA 04 / 08

Storage and databases

Study order: Know which storage fits which job; interviews ask this a lot.

01	**S3**	Store files of any size.
02	**S3 storage classes**	Cheaper tiers for old data.
03	**Bucket policies**	Who can read or write.
04	**Public bucket risks**	The classic data leak.
05	**EBS**	Disks for servers.
06	**EFS**	Shared file system.
07	**Snapshots**	Backups of disks.
08	**RDS**	Managed SQL databases.
09	**Multi-AZ**	A standby database in another zone.
10	**Read replicas**	Spread read traffic.
11	**DynamoDB**	Managed key-value database.
12	**ElastiCache**	Managed Redis.
13	**CloudFront**	CDN for fast static files.
14	**Lifecycle rules**	Move or delete old files automatically.
15	**Backups and retention**	How long to keep data.

Practice

1. Store user uploads safely and serve them fast. Design it.
2. Your RDS database is slow on reads. What do you try?

AREA 05 / 08

IAM and security

Study order: Most cloud breaches are bad permissions, not clever hackers.

01	**IAM users**	People who log in.
02	**IAM roles**	Permissions for services.
03	**Policies**	JSON rules for who can do what.
04	**Least privilege**	Only the access needed.
05	**MFA**	Protect every human login.
06	**Root account**	Lock it away.
07	**Access keys**	Rotate them, never commit them.
08	**VPC**	Your private network in the cloud.
09	**Public and private subnets**	What faces the internet.
10	**NAT gateway**	Private servers reaching out safely.
11	**Security groups vs NACLs**	Two layers of firewall.
12	**KMS**	Manage encryption keys.
13	**Secrets Manager**	Store passwords and keys.
14	**CloudTrail**	Log every API call.
15	**GuardDuty**	Detect suspicious activity.

Practice

1. A developer pushed an access key to GitHub. What now?
2. Design a VPC for a web app and a private database.

AREA 06 / 08

Terraform

Study order: Write infrastructure as code so you can rebuild everything in minutes.

01	**Infrastructure as code**	Servers defined in files.
02	**Providers**	Plug-ins for AWS, Azure and GCP.
03	**Resources**	The things Terraform creates.
04	**Variables**	Reuse the same code across environments.
05	**Outputs**	Values other code can use.
06	**State file**	Terraform's memory of what exists.
07	**Remote state**	Share state safely with a team.
08	**State locking**	Stop two people applying at once.
09	**plan and apply**	Preview, then change.
10	**Modules**	Reusable building blocks.
11	**Workspaces**	Separate dev and prod.
12	**Drift**	When reality differs from code.
13	**Import**	Bring existing resources under Terraform.
14	**CloudFormation**	AWS's own option.
15	**Terraform in CI**	Apply through pipelines, not laptops.

Practice

1. Write Terraform for an EC2 server behind a load balancer.
2. Two people ran apply at the same time. What breaks?

AREA 07 / 08

Containers and Kubernetes

Study order: Containers are how most cloud apps ship now.

01	**Docker basics**	Package an app with everything it needs.
02	**Dockerfile**	Build steps for an image.
03	**Image registry**	ECR or Docker Hub.
04	**Docker Compose**	Run several containers locally.
05	**ECS**	AWS's simpler container service.
06	**Fargate**	Containers without managing servers.
07	**Kubernetes basics**	Run containers at scale.
08	**Pods**	The smallest unit in Kubernetes.
09	**Deployments**	Roll out new versions.
10	**Services**	Stable address for pods.
11	**Ingress**	Route outside traffic in.
12	**ConfigMaps and Secrets**	Config outside the image.
13	**EKS**	Managed Kubernetes on AWS.
14	**Autoscaling pods**	More pods under load.
15	**Helm**	Package Kubernetes apps.

Practice

1. Deploy a containerised API on ECS or EKS.
2. A pod keeps restarting. How do you debug it?

AREA 08 / 08

Monitoring and cost

Study order: Companies hire cloud engineers to keep things running and bills sane.

01	**CloudWatch metrics**	CPU, memory and requests.
02	**CloudWatch logs**	All logs in one place.
03	**Alarms**	Get alerted before users notice.
04	**Dashboards**	One screen for system health.
05	**Tracing**	Follow one slow request.
06	**Uptime checks**	Is the site reachable?
07	**Incident response**	Detect, fix, write it up.
08	**Runbooks**	Steps for common failures.
09	**Cost Explorer**	See where money goes.
10	**Budgets and alerts**	Know before the bill jumps.
11	**Savings Plans**	Commit and pay less.
12	**Idle resources**	Delete what nobody uses.
13	**Data transfer costs**	The hidden part of the bill.
14	**CI/CD pipelines**	Ship changes safely.
15	**Interview prep**	Scenarios, not definitions.

Practice

1. Your AWS bill tripled this month. How do you find out why?
2. The site is down at 2 AM. Walk through your first 10 minutes.