

Extra

1) OS, শেল, টার্মিনাল

কী: Linux basics (Ubuntu), Shell (bash/zsh), Windows হলে WSL2, Terminal (Windows Terminal / iTerm2).

কেন: প্রায় সব সার্ভার Linux-এ—ফাইল, পারমিশন, প্রসেস, নেটওয়ার্ক কমান্ড না জানলে ডেপ্লয়/ডিবাগ আটকে ঘাবে।

শুরু:

```
ls -la; pwd; cd; cp; mv; rm -rf  
chmod 755 file; chown user:group file  
ps aux | grep node; kill -9 <pid>  
# নেটওয়ার্ক/ডায়াগ  
curl -v http://localhost:3000  
ping 8.8.8.8; netstat -tulpn | grep 3000
```

2) এডিটর/IDE

কী: VS Code (extensions: GitLens, ESLint, Prettier), IntelliJ IDEA (Java), PyCharm (Python).

কেন: দ্রুত রিফ্যাক্টরিং, ডিবাগার, টেস্ট রান—প্রোডাক্ষনভিত্তি বাঁচায়।

শুরু: Breakpoint দিয়ে ডিবাগ রান, format-on-save, tasks.json/launch.json কনফিগার করো।

3) Version Control (Git) + Hosted repos

কী: Git, GitHub/GitLab/Bitbucket.

কেন: টিমওয়ার্ক, কোড রিভিউ, CI/CD।

4) প্যাকেজ ম্যানেজার ও রানটাইম

JavaScript/TypeScript: Node.js (nvm), npm/yarn/pnpm

Python: pyenv, pip/pipx, Poetry/Pipenv

Java: SDKMAN!, Maven/Gradle

Others: Rust (rustup/cargo), Go (go modules), PHP (Composer)

কেন: ডিপেনডেন্সি, বিল্ড, স্ক্রিপ্ট, reproducible env.

শুরু: lockfile কমিট করো, .nvmrc/pyproject.toml রাখো।

5) Build টুল & টাক্স রানার

JS/TS: Vite, Webpack, esbuild

Java: Maven/Gradle

কেন: কোডকে প্রডাকশন-রেডি বানায় (bundle, minify, transpile)।

শুরু: "dev", "build", "test" স্ক্রিপ্ট নির্দিষ্ট করো।

6) API/ওয়েব ফ্রেমওয়ার্ক

Node.js: Express/Fastify/NestJS

Python: FastAPI/Django REST Framework

Java: Spring Boot

.NET: ASP.NET Core

কেন: REST/GraphQL সার্ভিস বানানো।

শুরু:

- রাউটিং, কন্ট্রোলার, সার্ভিস/রিপোজিটরি লেয়ার

- DTO/সকিমা ভ্যালিডেশন (zod/joi/pydantic)
- OpenAPI/Swagger ডক

7) ডেটাবেস & ORM

SQL: PostgreSQL (প্রথম পছন্দ), MySQL

NoSQL: MongoDB (document), Redis (cache), Elastic/OpenSearch (search)

ORM: Prisma (Node), TypeORM, Sequelize; Django ORM; SQLAlchemy;

JPA/Hibernate

কেন: ডাটা মডেল, ট্রানজ্যাকশন, পারফরম্যান্স।

শুরু:

- নরমালাইজেশন + ইনডেক্সিং
- মাইগ্রেশন (Prisma Migrate / Flyway / Alembic)
- কানেকশন পুলিং

8) টেস্টিং

Unit: Jest/Vitest (JS), pytest (Py), JUnit (Java)

Integration: Testcontainers (DB/queue সহ), supertest/pytest-httpx

E2E/UI: Playwright/Cypress

কেন: রিগ্রেশন ঠেকায়, রিফ্যাক্টরিং নিরাপদ।

শুরু: AAA প্যাটার্ন, deterministic টেস্ট ডাটা, coverage gate।

9) API ক্লায়েন্ট/ডিবাগ

কী: Postman/Insomnia, curl, HTTPie

কেন: দ্রুত এন্ডপয়েন্ট টেস্ট, স্ক্রিপ্টেবল রিকোয়েস্ট।

শুরু: Collections + environment vars + pre-request scripts।

10) কনটেইনার ও ভার্চুয়ালাইজেশন

কী: Docker, Docker Compose; Windows হলে Docker Desktop + WSL2

কেন: একই Env লোকাল/CI/প্রডে চালানো।

শুরু:

- ছোট base image (alpine/distroless)
- multi-stage build
- healthcheck, non-root user
- .dockerignore

```
FROM node:20-alpine AS build
WORKDIR /app
COPY package*.json .
RUN npm ci
COPY . .
RUN npm run build
FROM gcr.io/distroless/nodejs20
WORKDIR /app
COPY --from=build /app/dist ./dist
CMD ["dist/server.js"]
```

11) অর্কেস্ট্রেশন (বেসিক ধারণা)

কী: Kubernetes (kubectl, Helm), অথবা ছোট প্রজেক্টে Docker Compose/Swarm।

কেন: স্কেলিং, রোলআউট, secret/config আলাদা রাখা।

শুরুত: Deployment/Service/Ingress, liveness/readiness probes, Helm values।

12) CI/CD

কী: GitHub Actions, GitLab CI, Jenkins, CircleCI

কেন: অটো টেস্ট, বিল্ড, সিকিউরিটি স্ক্যান, ডেপলয়।

শুরুত:

- lint → test → build → scan → package → deploy স্টেজ
- ক্যাশিং ও artifacts
- environment-based secrets

13) ক্লাউড (একটা প্রোভাইডার গভীরে)

AWS (গুরুতে পছন্দ): EC2, S3, RDS, Lambda, API Gateway, CloudWatch, IAM, VPC

বিকল্প: GCP (GCE, GKE, Cloud Run, Cloud SQL), Azure (App Service, AKS, Cosmos)

IaC: Terraform/Pulumi—infra কোডে রাখো।

কেন: প্রতি রানটাইম বুবলে হবে—খরচ, সিকিউরিটি, স্কেল।

14) সিকিউরিটি বেসিন্স

- **Secrets:** .env নয়—AWS Secrets Manager/HashiCorp Vault/GitHub Secrets
- **Deps scan:** Dependabot, Snyk
- **Static analysis:** ESLint/Pylint + TypeScript/mypy + SonarQube
- **OWASP Top 10:** ইনপুট ভ্যালিডেশন, auth, rate-limit, CORS, HTTPS, headers
- **API hardening:** JWT/OAuth2, scopes, audit logs

15) অবজারভেবিলিটি

কী: লগ (ELK/EFK, Loki), metrics (Prometheus + Grafana), tracing (OpenTelemetry + Jaeger/Tempo)

কেন: প্রডে সমস্যা ধরার একমাত্র টেকসই উপায়।

শুরু: correlation/id, স্ট্রাকচার্ড লগ (JSON), RED/USE মেট্রিক্স।

16) নেটওয়ার্কিং ও পারফরম্যান্স

কী: cURL, ping, traceroute, nslookup/dig, Wireshark (গভীর ডিবাগ), nginx/Traefik (reverse proxy)

লোড টেস্ট: k6 / JMeter / Locust

ওয়েব পারফ: Lighthouse, Chrome DevTools (Performance/Network)

17) ডিপ্লয়মেন্ট ও সার্ভার অপস

কী: Nginx, systemd, PM2 (Node), process limits (ulimit), file descriptors

কনফিগ: 12-factor apps, env vars, config map/secret split

ব্লু-গ্রিন/ক্যানারি: downtime ছাড়া রিলিজ।

18) কোড কোয়ালিটি ও স্টাইল

Linters/Formatters: ESLint + Prettier, Black/ruff (Python)

Static type: TypeScript/mypy

Pre-commit hooks: husky + lint-staged / pre-commit (Python)

কেন: কোডবেস ক্লিন, রিভিউ দ্রুত।

19) ডকুমেন্টেশন ও নলেজ শেয়ারিং

কী: Markdown, ADRs (Architecture Decision Records),

Mermaid/PlantUML/C4 Model, Excalidraw/diagrams.net, Swagger/OpenAPI,

Docusaurus

কেন: টিমে context হারায় না; অনবোর্ডিং দ্রুত।

20) কল্যাব টুলস

কী: Jira/Trello/Linear, Notion/Confluence, Slack/Teams

কেন: প্ল্যানিং, ট্র্যাকিং, কমিউনিকেশন—ইঞ্জিনিয়ারিংও টিম স্পোর্ট।

ভূমিকা অনুযায়ী “অতিরিক্ত” (nice-to-have)

Frontend: React/Next.js, Tailwind, Zustand/Redux, React Query, Playwright; Vite; Web Vitals

Backend: Auth (OAuth2/OIDC), gRPC, GraphQL, message brokers (RabbitMQ/Kafka/SQS), CRON/queues (BullMQ/Celery), Nginx

Mobile: React Native/Flutter, Android Studio/Xcode, Fastlane

Data/ML: Jupyter, Pandas/NumPy, Airflow, Spark, MLflow, Dockerized notebooks

DevOps/SRE: Kubernetes গভীরে, Helm, ArgoCD, Prometheus Operator, Grafana, Alertmanager, Terraform modules

“Minimum Viable Toolbox” (আজ থেকেই শুরু করো)

- Windows + WSL2 (Ubuntu), VS Code
- Git + GitHub, Node.js (nvm) + npm/pnpm
- Express/FastAPI দিয়ে ছোট REST API
- PostgreSQL + Prisma/SQLAlchemy
- Jest/pytest (unit + simple integration)
- Docker + Docker Compose (DB, app, pgAdmin)
- GitHub Actions (lint → test → docker build)
- OpenAPI + Postman কালেকশন
- Basic logs + healthcheck; k6 দিয়ে mini লোড টেস্ট

বাস্তব অনুশীলন (৭টি মিনি-মাইলস্টোন)

1. CRUD REST API (users, auth, rate-limit)
2. DB মাইগ্রেশন + ইনডেক্স টিউনিং
3. ডকারাইজ + compose (app+db+adminer/pgadmin)
4. GitHub Actions pipeline + Dependabot
5. Observability: structured logs, request id, basic metrics
6. E2E: Playwright/Cypress দিয়ে ২-৩টা happy path টেস্ট
7. Cloud ডেপ্লয়: ১টি VM-এ Nginx reverse proxy বা Render/Fly/ Railway-তে container deploy

ছোট টিপস (যা “ভালো” ইঞ্জিনিয়ারদের আলাদা করে)

- Automation first: রিপিটিটিভ কিছু দেখলে স্ক্রিপ্ট বানাও।
- Security by default: secret কখনো রিপোতে নয়; লিস্ট প্রিভিলেজ।
- Observability habit: প্রতিটি ফিচারে log+metric ঘোগ করো।
- Docs as you code: README, ADR, OpenAPI সর্বদা আপডেট।
- Review discipline: ছোট PR, স্পষ্ট ডিফ, টেস্ট সহ।



Advanced Tools & Software for Software Engineers

1. Advanced Version Control & Collaboration

- GitHub Actions / GitLab CI/CD → অটো টেস্ট, ডকার বিল্ড, ডেপ্লয়।
- Gerrit / Phabricator → বড় টিমে কোড রিভিউ টুল।
- SonarQube → কোড কোয়ালিটি, সিকিউরিটি হোল ধরতে।

2. DevOps & Infrastructure

- Terraform / Pulumi → Infrastructure as Code (AWS, GCP, Azure automate).
- Ansible / Chef / Puppet → সার্ভার কনফিগারেশন অটোমেশন।
- Consul / Vault → সার্ভিস ডিসকভারি ও সিক্রেট ম্যানেজমেন্ট।
- Kubernetes (kubectl, Helm, ArgoCD) → কনটেইনার orchestration ও GitOps ডেপ্লয়।
- Istio / Linkerd → Service Mesh (zero-trust, traffic routing, observability)।

3. Observability & Monitoring

- Prometheus + Grafana → Metrics collection & visualization।
- ELK Stack (Elasticsearch, Logstash, Kibana) / Loki → লগ ম্যানেজমেন্ট।
- Jaeger / Zipkin / OpenTelemetry → Distributed Tracing (microservices debug)।
- Datadog / New Relic / Sentry → SaaS observability ও error tracking।

4. Performance & Load Testing

- k6 / JMeter / Locust → Load, stress, spike testing।
- wrk / ab (Apache Benchmark) → ছোট scale পারফরম্যান্স টেস্ট।
- Flamegraph (perf, py-spy, 0x) → কোড/সিস্টেম প্রোফাইলিং।

5. Security Engineering

- OWASP ZAP / Burp Suite → পেন-টেস্টিং টুলস।
- Trivy / Clair / Anchore → Docker image vulnerability scan।
- Snyk / Dependabot / Whitesource → Dependency security check।
- HashiCorp Vault / AWS Secrets Manager → সিক্রেট ম্যানেজমেন্ট।
- Keycloak / Auth0 → OAuth2 / SSO authentication system।

6. Databases & Data Engineering

- Advanced SQL Tools:
 - pgAdmin / DBeaver / DataGrip → GUI SQL client।
 - pgBouncer / ProxySQL → DB connection pooling।
- Caching & Queues:
 - Redis / Memcached → cache।
 - RabbitMQ / Kafka / AWS SQS → asynchronous processing, streaming।
- ETL & Data Tools:
 - Airflow / Dagster / Prefect → Data pipeline orchestration।
 - dbt → SQL transformation।

7. API & Integration

- GraphQL (Apollo, Hasura) → Modern API design।
- gRPC / Protobuf / Thrift → High performance service communication।

- Kong / NGINX Ingress / Traefik → API Gateway / reverse proxy |
- Postman / Insomnia / Hoppscotch → Advanced API testing |

8. Frontend (Advanced UI/UX Tooling)

- React + Next.js / Vue + Nuxt / Angular → Production-ready SPA/SSR |
- Storybook → UI component library & visual testing |
- Playwright / Cypress → E2E UI Testing |
- Webpack / Vite / Turborepo / Nx → Monorepo build tool |

9. Collaboration & Documentation

- Notion / Confluence → জ্ঞান শেয়ারিং, ডকুমেন্টেশন |
- Jira / Linear → প্রজেক্ট ম্যানেজমেন্ট |
- Miro / Excalidraw / Lucidchart → Architecture diagram & workflow design |
- C4 Model Tools / PlantUML / Mermaid → সিস্টেম আর্কিটেকচার ডকুমেন্টেশন |

10. Cloud & Serverless

- AWS Lambda / GCP Cloud Functions / Azure Functions → Serverless computing |

- AWS Fargate / Cloud Run → container serverless runtime।
- API Gateway + Lambda + DynamoDB (Serverless Stack) → আধুনিক scalable architecture।
- Pulumi / Serverless Framework → IaC for serverless apps।

11. Machine Learning & Data Science (যদি কাজ করো ML/DL-এ)

- Jupyter Notebook / Google Colab → স্বত্ত্বালন মডেল প্রোটোটাইপ।
- Pandas / NumPy / Spark → ডেটা প্রসেসিং।
- MLflow / Kubeflow → ML model training pipeline।
- TensorFlow / PyTorch → Deep Learning framework।

12. Miscellaneous Advanced Tools

- Helmfile / Kustomize → Kubernetes manifest management।
- Caddy / Envoy Proxy → Modern reverse proxy।
- gitleaks → git history থেকে secret leak ধরার টুল।
- fzf / bat / ripgrep / tmux → Advanced terminal productivity।
- Nix / Guix → Reproducible development environments।

- ✓ সহজভাবে বললে—
- 👉 শুরুতে Core Tools (Git, Docker, SQL, API, Testing, CI/CD) এ মাস্টারি।
- 👉 এরপর Advanced Tools (Terraform, Kubernetes, Prometheus, Kafka, Istio, Vault) হাতে কলমে শিখো।
- 👉 সবশেষে Cloud & Observability-তে ভালো ধারণা রাখো—তাহলেই তুমি Industry Expert! 