## Simple Prompts

1. **Disk‑usage alert script**  
   “Write a Bash script that checks disk usage every 5 minutes and sends an email if usage exceeds 80%.”
2. **Service‑restart health check**  
   “Generate a PowerShell script to verify that the ‘W3SVC’ service is running, and restart it if it is stopped.”
3. **Log‑rotation cron job**  
   “Provide a crontab entry and shell script to rotate /var/log/nginx/\*.log daily and keep the last 7 archives.”
4. **NTP configuration**  
   “Write an Ansible task to install and configure ntp on Ubuntu 20.04, pointing to time.google.com.”
5. **User‑creation playbook**  
   “Generate an Ansible playbook to create a Linux user deploy with sudo privileges and an SSH key.”
6. **Docker‑build command**  
   “Show the Docker CLI command to build an image from the current directory, tag it as myapp:latest, and push to Docker Hub.”
7. **Simple Terraform module**  
   “Write a Terraform module that provisions a single AWS EC2 t3.micro instance in us‑east‑1.”
8. **Systemd unit file**  
   “Generate a systemd unit file for a Python Flask app located at /opt/app/app.py.”
9. **Environment‑variable export**  
   “Provide a shell snippet to load environment variables from a .env file into the current session.”
10. **Basic Kubernetes deployment**  
    “Write a Kubernetes Deployment YAML for Nginx with two replicas and a LoadBalancer Service.”

## Medium Prompts

1. **Idempotent Ansible role**  
   “Create an Ansible role nginx that installs Nginx, ensures the service is enabled, and deploys a custom nginx.conf template.”
2. **Parameterized Terraform VPC**  
   “Write a Terraform configuration with variables for cidr\_block, public\_subnets, and tags to create a VPC and public subnets.”
3. **Jenkins Pipeline**  
   “Generate a declarative Jenkinsfile that checks out Git, builds a Maven project, runs unit tests, and archives artifacts.”
4. **Blue‑Green deploy in Kubernetes**  
   “Provide a Helm chart snippet that implements blue‑green deployment for a Node.js app, with separate services for v1 and v2.”
5. **AWS Auto Scaling group**  
   “Write Terraform code to create an AWS Auto Scaling group behind an ALB, scaling between 2 and 5 t3.medium instances.”
6. **SSL cert issuance**  
   “Generate an Ansible playbook to obtain and renew Let’s Encrypt certificates using certbot for multiple domains.”
7. **Vault Secrets injection**  
   “Write a Kubernetes mutating webhook configuration that injects HashiCorp Vault secrets into pods via annotations.”
8. **CI code‑coverage report**  
   “Produce a GitLab CI snippet that runs pytest with coverage, fails if coverage < 80%, and uploads the report.”
9. **Logging stack with ELK**  
   “Create a Docker Compose file to deploy Elasticsearch, Logstash, and Kibana, with environment variables for basic auth.”
10. **Azure ARM template**  
    “Write an Azure Resource Manager template to provision an App Service Plan and a Web App with application settings.”

## Advanced Prompts

Use APE/RACE/COAST frameworks, specify format, few‑shots, and TDD style wherever possible.

1. **[APE] Automated Terraform Review**

[Action]: Review Terraform code in this repo for AWS best practices.

[Purpose]: Ensure security, cost‑efficiency, and modularity.

[Expectation]: Provide a Markdown checklist of issues and suggested fixes.

1. **[RACE] GitOps CI/CD pipeline**

[Role]: As a DevOps engineer using GitLab CI.

[Action]: Generate a `.gitlab-ci.yml` that builds Docker, pushes to registry, and deploys via ArgoCD.

[Context]: Project uses Helm charts in `charts/`.

[Expectation]: Include stages: lint, build, push, deploy with retry logic.

1. **COAST‑style Kubernetes operator**

[Context]: You are a Site Reliability Engineer.

[Objective]: Automate database schema migrations in Kubernetes.

[Actions]: Write Go code for a custom operator using Kubebuilder.

[Scenario]: Trigger migrations on Deployment rollout.

[Task]: Include CRD, controller logic, and example manifests.

1. **TDD Jenkins shared library**  
   “Write a Jenkins shared library function notifySlack(channel, message) in Groovy with unit tests (using JenkinsPipelineUnit) that sends a Slack notification via webhook. Include test cases for success and failure scenarios.”
2. **Ansible Molecule testing**  
   “Generate an Ansible role postgresql with Molecule tests on Docker for Ubuntu and CentOS, verifying service, port, and user creation.”
3. **Terraform policy as code**  
   “Produce a Sentinel policy that denies AWS security groups with 0.0.0.0/0 ingress, and include unit tests using sentinel test format.”
4. **Agentic AIOps workflow**  
   “Design a LangChain agent that ingests Prometheus metrics, diagnoses anomalies, and triggers remediation via the Kubernetes API. Include Python classes and sample conversation.”
5. **Immutable server build**  
   “Using Packer, create an image for AWS AMI with Docker, Node.js, and a healthcheck.sh script. Provide JSON template and post‑processor to register the AMI.”
6. **Prometheus custom exporter**  
   “Write a Python Prometheus exporter exposing CPU/memory of a custom process with /metrics endpoint, include setup.py and example systemd unit.”
7. **Azure DevOps multi‑stage pipeline**  
   “Generate an Azure Pipelines YAML that builds .NET Core, runs SonarQube scan, publishes to Azure Artifacts, and deploys to an AKS namespace with Helm, parameterized by environment.”

## 10 Complex (Chained & Agentic) Prompts

Demonstrate prompt‑chaining, tree‑of‑thoughts, and self‑consistency for end‑to‑end infra automation.

1. **Multi‑phase GitOps rollout**

#1 “Generate an ArgoCD Application YAML for prod and staging.”

#2 “Validate the manifest against Kubernetes 1.21 schema.”

#3 “Write a rollback script that reverts an Application to the previous healthy commit.”

Chain these into one agentic prompt that outputs each artifact in order.

1. **Self‑healing cluster blueprint**  
   “Using Tree‑of‑Thoughts, outline steps to detect node failures in a Kubernetes cluster, cordon/drain them, spin up replacements, and update DNS. Then generate Terraform, Ansible, and Helm code for each step with self‑consistency checks.”
2. **Full‑stack AIOps platform**  
   “Assemble a microservice architecture: ingest logs via Fluentd, store in Loki, visualize in Grafana, automate alerts with LLM‑driven responders in Python. Produce Docker Compose, service code stubs, and a README guiding multi‑stage deployment.”
3. **ChatOps bot generator**  
   “Create a Botpress configuration that listens in Slack for /deploy feature‑branch, clones the repo, builds Docker, pushes to ECR, updates a Kubernetes Deployment, and reports back success or failure.”
4. **Hybrid‑cloud failover**  
   “Write a Python script leveraging boto3 and google‑cloud SDKs to replicate S3 objects to GCS, monitor replication lag, and failover DNS in Route 53 based on health checks. Include retries and exponential backoff.”
5. **Security‑hardened CI/CD**  
   “Generate a Jenkins Declarative Pipeline that integrates Trivy, Checkov, and OWASP ZAP, fails on high‑severity findings, and posts a summary to Microsoft Teams via webhook.”
6. **Multi‑tenant monitoring stack**  
   “Compose Terraform and Helm charts to deploy Prometheus Operator, Thanos for long‑term storage, and Grafana multitenancy with LDAP auth. Include sample values for two tenants and an upgrade path.”
7. **Chaos Engineering with AI**  
   “Using COAST, craft a LitmusChaos experiment YAML to randomly kill 10% of pods, monitor via Dynatrace API, and invoke an AI agent to analyze the incident and propose resilience improvements.”
8. **Infrastructure drift remediation**  
   “Develop a Golang CLI that scans live AWS resources, compares with Terraform state via terraform show, and auto‑generates Terraform imports or drift‑fix patches. Include unit tests using Go’s testing package.”
9. **End‑to‑end observability AI agent**  
   “Create a LangChain agent blueprint that uses OpenAI to parse PromQL queries from natural language (e.g., ‘show me CPU spikes in the last hour’), executes them against Prometheus, and visualizes results in Grafana via API, all with error handling and retry logic.”

Feel free to tweak each prompt’s **temperature**, **few‑shot examples**, and **model** parameters to fine‑tune creativity vs. determinism.

**1. Simple Prompts**

*Straightforward tasks with minimal context or constraints.*

1. **Action**: "Write a Bash script to restart a Kubernetes pod."
2. **Action**: "Generate a Terraform template to create an AWS S3 bucket."
3. **Action**: "List 5 common edge cases for testing a CI/CD pipeline."
4. **Action**: "Draft a YAML file to deploy a Docker container on ECS."
5. **Action**: "Suggest 3 commands to check disk usage on a Linux server."
6. **Action**: "Create a Python script to ping a server and log response time."
7. **Action**: "Write a cron job to automate log cleanup every 24 hours."
8. **Action**: "Generate test data for a load balancer health check."
9. **Action**: "Provide a basic Ansible playbook to install Nginx."
10. **Action**: "List 3 security best practices for AWS IAM roles."

**2. Medium Prompts**

*Detailed tasks with specific tools or scenarios.*

1. **RACE Framework**:
   * **[Role]**: As a DevOps engineer using AWS and Terraform.
   * **[Action]**: Write a Terraform module to provision an auto-scaling group.
   * **[Context]**: The group must use t3.medium instances and scale based on CPU usage.
   * **[Expectation]**: Include security group rules for HTTP/HTTPS traffic.
2. **APE Framework**:
   * **[Action]**: Generate a Python script to monitor EC2 instance status.
   * **[Purpose]**: To alert the team via Slack if instances are unhealthy.
   * **[Expectation]**: Use Boto3 and filter instances tagged as production.
3. **TAG Framework**:
   * **[Task]**: Create a test case for a failed Kubernetes deployment rollback.
   * **[Action]**: Simulate a deployment failure and validate rollback steps.
   * **[Goal]**: Ensure the system reverts to the last stable version.
4. **Context**: "You are troubleshooting a slow API. Write a curl command to test latency and include headers for authentication."
5. **Action**: "Using Python, write a script to parse Jenkins logs and extract failed build IDs."
6. **Action**: "Design a Grafana dashboard to monitor AWS Lambda cold starts."
7. **Action**: "Generate a Helm chart for deploying Redis with persistence enabled."
8. **Action**: "Create a PowerShell script to automate Azure VM backups every 6 hours."
9. **Action**: "Write a Prometheus alert rule to detect high memory usage in a Node.js app."
10. **Action**: "Suggest 5 test scenarios for validating a blue-green deployment strategy."

**3. Advanced Prompts**

*Multi-step tasks with integration or predictive elements.*

1. **COAST Framework**:
   * **[Context]**: A microservice is experiencing intermittent downtime.
   * **[Objective]**: Identify root causes and automate remediation.
   * **[Actions]**:
     + Use Prometheus to scrape metrics.
     + Write a Python script to analyze logs for 5xx errors.
     + Trigger a Lambda function to restart the service if errors exceed 10%.
2. **CRISPE Framework**:
   * **[Clarity]**: Optimize a Kubernetes cluster for cost efficiency.
   * **[Relevance]**: Use Kubecost and Terraform.
   * **[Specificity]**: Downscale non-production namespaces during off-hours.
3. **RELIC Framework**:
   * **[Role]**: Act as a Site Reliability Engineer (SRE).
   * **[Emphasis]**: Reduce AWS costs by 20%.
   * **[Information]**: Current usage: 100 t3.large instances, 50% idle.
   * **[Challenge]**: Design an auto-scaling policy with Spot Instances.
4. **Action**: "Train a linear regression model to predict EC2 instance demand based on historical CPU/RAM usage (provide sample CSV data)."
5. **Action**: "Integrate OpenAI’s API with PagerDuty to auto-generate incident summaries from alerts."
6. **Action**: "Write a Python script using the Isolation Forest algorithm to detect anomalous network traffic in VPC flow logs."
7. **Action**: "Build a ChatGPT-powered chatbot to answer FAQs about AWS outage history (sync with AWS Trust Advisor)."
8. **Action**: "Create a Jenkins pipeline that uses AI to prioritize failed tests based on historical flakiness data."
9. **Action**: "Design a self-healing pipeline: If a deployment fails, roll back and trigger a Chaos Monkey test."
10. **Action**: "Use LangChain to generate Terraform code from natural language prompts (e.g., ‘Deploy a secure EKS cluster’)."

**4. Complex Prompts**

*Scenario-based tasks requiring cross-tool integration or ethical governance.*

1. **SCAMPER Framework**:
   * **[Substitute]**: Replace manual security audits with AI-driven vulnerability scanning in CI/CD.
   * **[Combine]**: Merge Prometheus, Grafana, and GPT-4 to auto-generate incident post-mortems.
   * **[Reverse]**: Simulate a ransomware attack and design a recovery playbook.
2. **Six Thinking Hats Framework**:
   * **[White Hat]**: Analyze metrics from 10 production outages.
   * **[Red Hat]**: Predict team morale impact if incidents increase by 30%.
   * **[Green Hat]**: Propose an AI tool to reduce MTTR by 50%.
3. **BAB Framework**:
   * **[Before]**: Manual capacity planning causes 40% overprovisioning.
   * **[After]**: AI predicts resource needs with 95% accuracy.
   * **[Bridge]**: Build a model using historical Kubernetes cluster data.
4. **Action**: "Orchestrate a multi-cloud disaster recovery plan using Azure, AWS, and GPT-4 for automated failover decision-making."
5. **Action**: "Develop an ethical AI governance policy for autonomous Kubernetes scaling (include bias checks and cost-transparency reports)."
6. **Action**: "Create a self-optimizing CI/CD pipeline that uses reinforcement learning to balance speed vs. stability."
7. **Action**: "Design a ChatGPT plugin to audit Terraform code for GDPR compliance (e.g., data residency checks)."
8. **Action**: "Simulate a DDoS attack on AWS infrastructure and auto-generate mitigation steps using AI-driven playbooks."
9. **Action**: "Build a closed-loop system where AI analyzes post-incident reviews to update runbooks and training materials."
10. **Action**: "Integrate LLMs with Istio to dynamically adjust traffic routing based on real-time sentiment analysis of user logs."

**Key Takeaways**

* **Frameworks Matter**: Use RACE, CRISPE, or RELIC to structure prompts for clarity.
* **Tool Integration**: Pair prompts with tools like Terraform, Prometheus, or GPT-4.
* **Ethics & Governance**: Include checks for bias, cost, and compliance in complex scenarios.