Cloud Computing Mini Project Ideas PDF

A cloud computing mini project should teach one service or practice in one to four weeks: hosting a static site, a serverless API, a Docker container deployment, a simple CI/CD pipeline or monitoring with alerts. Use free tiers, set a billing alert first, and delete resources when done.

Key takeaways

- Start with static hosting and serverless functions; they cost little or nothing.
- Learn Docker early; it is used in almost every cloud job.
- Set a billing alert before you create anything.
- Take screenshots as you go for the report.
- Each idea fits in one to four weeks.

Who this PDF is for

This PDF is for 2nd and 3rd year students who want hands-on cloud practice. Each idea uses one or two services and can be done on AWS, Azure or Google Cloud free tiers, or locally with Docker. Plans list services, steps and viva points.

How to choose your project

- **One provider**, ideally the one your course uses.
- **Small resources** only: the smallest instance sizes and short run times.
- **Document as you click**: screenshots and commands.
- **Tear down** at the end and note it in the report.

Beginner mini projects (1 to 2 weeks)

1. Static Website on Cloud Storage

- **Problem:** Host a portfolio site on object storage with HTTPS.
- **Cloud services:** AWS S3 and CloudFront, or Azure Static Web Apps.
- **Tech stack:** HTML, CSS, AWS CLI.
- **Main modules:** Bucket setup, upload, CDN, custom error page.
- **Steps:** Create the bucket, upload files, put a CDN in front, test from a phone.
- **Show in the viva:** Why a CDN makes sites faster.

- **Extensions:** Add a custom domain.

2. Launch and Secure a Virtual Machine

- **Problem:** Run a web server on a cloud VM safely.
- **Cloud services:** AWS EC2 or Azure VM (free-tier size).
- **Tech stack:** Linux, Nginx, SSH keys.
- **Main modules:** VM launch, key login, firewall rules, Nginx install, snapshot.
- **Steps:** Launch with SSH keys, open only ports 22 and 80, install Nginx, take a snapshot.
- **Show in the viva:** Security groups in simple words.
- **Extensions:** Restrict SSH to your IP.

3. Serverless Contact Form

- **Problem:** Save form messages without a server.
- **Cloud services:** AWS Lambda, API Gateway, DynamoDB (or Azure Functions).
- **Tech stack:** JavaScript, Python or Node.js.
- **Main modules:** Form page, API endpoint, function, database table.
- **Steps:** Write the function, expose it with API Gateway, save to DynamoDB, call from the form.
- **Show in the viva:** How the function runs only when called.
- **Extensions:** Send an email notification.

4. Dockerise a Web App

- **Problem:** Package an app into a container.
- **Cloud services:** Docker Desktop on your laptop; Docker Hub.
- **Tech stack:** Docker, any small web app.
- **Main modules:** Dockerfile, image build, run, push to registry.
- **Steps:** Write a Dockerfile, build, run locally, push to Docker Hub.
- **Show in the viva:** Images versus containers.
- **Extensions:** Use a multi-stage build to shrink the image.

Intermediate mini projects (2 to 3 weeks)

5. Docker Compose Multi-container App

- **Problem:** Run an app with its database in containers.
- **Cloud services:** Docker and Docker Compose locally.
- **Tech stack:** Node.js or Python app, PostgreSQL or MongoDB.
- **Main modules:** Compose file, networks, volumes, environment variables.
- **Steps:** Define services, connect app to database, persist data with volumes.
- **Show in the viva:** Why volumes keep data after restart.
- **Extensions:** Add an Nginx reverse proxy.

6. CI/CD with GitHub Actions

- **Problem:** Test and deploy automatically on each push.
- **Cloud services:** GitHub Actions; a free hosting target.

- **Tech stack:** YAML, any app with tests.
- **Main modules:** Test job, build job, deploy job, status badge.
- **Steps:** Write the workflow, run tests, deploy on success.
- **Show in the viva:** What happens when a test fails.
- **Extensions:** Add a linting step.

7. Cloud Monitoring and Alerts

- **Problem:** Get an alert when a server is overloaded.
- **Cloud services:** AWS CloudWatch or Azure Monitor; a small VM.
- **Tech stack:** CloudWatch alarms, SNS email.
- **Main modules:** Metrics, dashboard, alarm, notification, stress test.
- **Steps:** Create a dashboard, set a CPU alarm, run a stress test, receive the email.
- **Show in the viva:** Metrics, logs and alarms.
- **Extensions:** Add log-based alerts.

8. Automated Backups to Cloud Storage

- **Problem:** Back up a folder or database to the cloud daily.
- **Cloud services:** AWS S3 with lifecycle rules, or Azure Blob.
- **Tech stack:** Bash or Python, cron, AWS CLI.
- **Main modules:** Backup script, schedule, lifecycle to cheaper storage, restore test.
- **Steps:** Write the script, schedule with cron, set lifecycle rules, test a restore.
- **Show in the viva:** Why a backup is useless without a restore test.
- **Extensions:** Encrypt backups.

Stretch mini projects (3 to 4 weeks)

9. Infrastructure as Code with Terraform

- **Problem:** Create a VM, network and bucket with code.
- **Cloud services:** AWS or Azure free tier.
- **Tech stack:** Terraform.
- **Main modules:** Provider setup, variables, resources, outputs, destroy.
- **Steps:** Write Terraform files, plan, apply, change one setting, destroy.
- **Show in the viva:** What terraform plan shows.
- **Extensions:** Store state remotely.

10. Kubernetes on Your Laptop

- **Problem:** Deploy and scale an app on a local cluster.
- **Cloud services:** minikube or kind locally.
- **Tech stack:** Docker, kubectl, YAML.
- **Main modules:** Deployment, service, scaling, rolling update.
- **Steps:** Deploy your image, expose it, scale replicas, roll out a new version.
- **Show in the viva:** Pods and deployments.
- **Extensions:** Add a Horizontal Pod Autoscaler.

11. Serverless URL Shortener

- **Problem:** Shorten links using only managed services.
- **Cloud services:** AWS Lambda, API Gateway, DynamoDB.
- **Tech stack:** Python or Node.js.
- **Main modules:** Create and redirect endpoints, storage, click counts.
- **Steps:** Store code-to-URL items, redirect with a 301, count clicks.
- **Show in the viva:** How DynamoDB keys are chosen.
- **Extensions:** Add expiry with DynamoDB TTL.

12. Chatbot on Cloud Functions

- **Problem:** Answer FAQs through a messaging bot.
- **Cloud services:** A cloud function plus Telegram Bot API.
- **Tech stack:** Python, webhooks.
- **Main modules:** Webhook endpoint, FAQ matching, logs.
- **Steps:** Create the bot, point its webhook to your function, reply from an FAQ list.
- **Show in the viva:** How webhooks work.
- **Extensions:** Store chat logs in a database.

Mini project report structure

A mini project report is usually 10 to 20 pages. Keep it short and clear:

1. **Title page and certificate** in your department's format.
2. **Introduction:** the problem in one paragraph and your objectives as a list.
3. **Tools used:** cloud provider and services, CLI or console, Docker or Terraform versions.
4. **Design:** one block diagram or flowchart, plus a simple architecture diagram.
5. **Implementation:** what each module does, with 3 to 6 screenshots.
6. **Results:** screenshots of running resources, test results and proof that resources were deleted.
7. **Conclusion and future scope,** then references.

Ask your guide for the exact format before you start writing.

Common mistakes to avoid

- No billing alert and resources left running.
- Using root credentials or putting access keys in code.
- Opening all ports to the internet.
- No architecture diagram in the report.

Frequently asked questions

What is a good cloud computing mini project for beginners?

Hosting a static website on cloud storage or a serverless contact form. Both use free-tier services and teach core ideas.

Do I need a credit card for cloud free tiers?

Most providers ask for a payment method on sign-up. Set a billing alert and use small resources to avoid charges.

Should I learn Docker for cloud projects?

Yes. Docker is used widely in cloud and DevOps work, and it runs free on your laptop.

Which cloud provider is best for students?

Use the one your course or target certification uses. All three major providers have student or free-tier offers.

Does this PDF include full source code?

No. It gives the problem, data, stack, modules and steps for each project. You write the code yourself, which is what examiners test in the viva.

Is this PDF free?

Yes. Fill in the short form on this page and the PDF downloads right away.

Sources and further reading

- AWS Free Tier
- Docker documentation
- GitHub Actions documentation
- Terraform documentation

About this PDF

Prepared by the Uncodemy training team as practice project ideas. Datasets and tools named are public; check each dataset's licence before use. We update it every January. Corrections: info@uncodemy.com.

More free resources and courses: uncodemy.com | info@uncodemy.com | +91 98183 66550