

Task	Description	Example tools
Cost optimization	Pick proper Instance types, use spot and reserved Instances, use auto scaling, and clean up unused resources.	Auto scaling, Infracost
Documentation	Document your code, architecture, and practices. Create playbooks to respond to incidents.	READMEs, wikis, Slack, IaC
Tests	Write automated tests for your infrastructure code. Run tests after every commit and nightly.	Terratest, tflint, OPA, InSpec

Most developers are aware of the first few tasks: install, configure, provision, and deploy. It's all the ones that come after them that catch people off guard. For example, did you think through the resilience of your service and what happens if a server goes down? Or a load balancer goes down? Or an entire datacenter goes dark? Networking tasks are also notoriously tricky: setting up VPCs, VPNs, service discovery, and SSH access are all essential tasks that can take months and yet are often entirely left out of many project plans and time estimates. Security tasks, such as encrypting data in transit using TLS, dealing with authentication, and figuring out how to store secrets, are also often forgotten until the last minute.

Every time you're working on a new piece of infrastructure, go through this checklist. Not every single piece of infrastructure needs every single item on the list, but you should consciously and explicitly document which items you've implemented, which ones you've decided to skip, and why.

Production-Grade Infrastructure Modules

Now that you know the list of tasks that you need to do for each piece of infrastructure, let's talk about the best practices for building reusable modules to implement these tasks. Here are the topics I'll cover:

- Small modules
- Composable modules