Here’s an 80-hour, 10-day bootcamp plan—8 hours per day—designed to take ten engineers from Python fundamentals through a full Agentic-AI + GCP/Vertex/Kubernetes/Dialog Flow stack. After the bootcamp, each team member will own a small project to cement their learning.

## **1. 10-Day Schedule (80 Hours)**

| **Day** | **Module** | **Hours** |
| --- | --- | --- |
| 1 | **Python Refresher & GCP Core Services** | 4 h + 4 h |
| 2 | **GCP Deep Dive (Compute, Storage, IAM)** | 8 h |
| 3 | **Vertex AI Fundamentals** | 8 h |
| 4 | **Vertex AI Advanced & Pipelines** | 8 h |
| 5 | **Kubernetes Essentials** | 8 h |
| 6 | **Kubernetes + Vertex AI on K8s** | 8 h |
| 7 | **Dialog Flow Conversational Agents** | 8 h |
| 8 | **Agentic AI Concepts & LLM Overview** | 8 h |
| 9 | **LangChain & LangGraph Deep Dive** | 8 h |
| 10 | **Capstone Sprint & Presentations** | 8 h |

## **2. Detailed Module Breakdown**

### **Day 1 (8 h)**

1. **Python Refresher (4 h)**
   * Advanced data structures (deque, defaultdict, pathlib…)
   * Asynchronous I/O with asyncio
   * Packaging & virtual environments (venv, Poetry)
   * Hands-on: Build a small CLI tool that reads/writes JSON
2. **GCP Core Services (4 h)**
   * GCP project setup, billing, IAM best practices
   * Networking basics: VPC, subnets, firewall rules
   * Storage options: Cloud Storage vs. Filestore vs. BigQuery
   * Lab: Provision a GCP project, create service accounts, and spin up a Storage bucket

### **Day 2 (8 h) – GCP Deep Dive**

* **Compute**: Compute Engine vs. GKE vs. Cloud Run
* **Data Services**: BigQuery, Pub/Sub, Cloud SQL
* **IAM & Security**: Roles, service accounts, workload identity
* **Lab**:  
  1. Deploy a container to Cloud Run
  2. Publish/subscribe messages via Pub/Sub
  3. Query a sample dataset in BigQuery

### **Day 3 (8 h) – Vertex AI Fundamentals**

* **Vertex AI Concepts**: Datasets, training vs. tuning, batch vs. online prediction
* **Managed Notebooks & AutoML**
* **Training custom models** with gcloud ai custom-jobs create
* **Lab**:  
  1. Ingest a CSV dataset
  2. Train an AutoML Tabular model
  3. Deploy for online prediction

### **Day 4 (8 h) – Vertex AI Advanced**

* **Hyperparameter tuning** & Vizier
* **Pipelines**: Kube Flow Pipelines on Vertex
* **Model monitoring & explanation**
* **Lab**:  
  1. Build a Vertex Pipelines workflow (data → train → deploy)
  2. Add a drift-monitoring component

### **Day 5 (8 h) – Kubernetes Essentials**

* **K8s Core Concepts**: Pods, Deployments, Services, ConfigMaps, Secrets
* **Helm**: Chart basics
* **Namespaces & RBAC**
* **Lab**:  
  1. Deploy a simple FastAPI app
  2. Expose it via a Service + Ingress
  3. Package as a Helm chart

### **Day 6 (8 h) – Kubernetes + Vertex AI Integration**

* **Custom K8s Operators** for Vertex AI
* **TFJob/Katib** for on-cluster training & tuning
* **GitOps** with Argo CD
* **Lab**:  
  1. Train a model on-cluster via a TFJob
  2. Automate pipeline deployment with Argo

### **Day 7 (8 h) – Dialog Flow Conversational Agents**

* **Dialog Flow CX vs. ES**
* **Intents, Entities, Webhooks**
* **Slot-filling & Contexts**
* **Lab**:  
  1. Build a multi-intent support bot
  2. Integrate with a webhook running on Cloud Run

### **Day 8 (8 h) – Agentic AI & LLM Overview**

* **Agentic AI paradigms**: single vs. multi-agent, supervisor patterns
* **LLM fundamentals**: embeddings, tokenization, context windows
* **Fine-tuning vs. prompt engineering**
* **Lab**:  
  1. Prompt-engineer a GPT-style model via Vertex Text API
  2. Compare responses with and without fine-tuning

### **Day 9 (8 h) – LangChain & LangGraph Deep Dive**

* **LangChain**: Chains, Agents, Callbacks, Memory
* **LangGraph Supervisor framework**
* **Multi-modal connectors** (PDFs, databases, APIs)
* **Lab**:  
  1. Build a RAG pipeline ingesting docs from Cloud Storage
  2. Orchestrate two specialized agents via LangGraph

### **Day 10 (8 h) – Capstone Sprint & Presentations**

* **Sprint**: Teams break into pairs; implement end-to-end “mini-project” using any subset of learned tools (e.g. Dialog Flow → LangChain → Vertex AI → Kubernetes).
* **Presentations**: 15 min per pair—architecture, demo, lessons learned.

## **3. Post-Training Individual Assignments**

| **Member** | **Focus Area** | **Project Deliverable** |
| --- | --- | --- |
| 1 | **GCP Infrastructure** | IaC templates (Terraform) provisioning VPC, IAM, Storage, GKE cluster |
| 2 | **Vertex AI Model Development** | End-to-end pipeline: data ingestion, training, deployment, monitoring |
| 3 | **Kubernetes Operations** | CI/CD for containerized FastAPI + Helm chart + Argo CD integration |
| 4 | **Dialog Flow Agent** | Multi-intents support bot with slot-filling, contexts, custom webhook integration |
| 5 | **Agentic AI Architecture** | Design & implement a two-agent LangGraph workflow with supervisor pattern |
| 6 | **LLM Fine-Tuning** | Fine-tune an open-source LLM on domain data, deploy via Vertex Text endpoint |
| 7 | **LangChain RAG Pipeline** | Build a PDF+DB ingestion pipeline, answer queries with context-aware retrieval |
| 8 | **LangGraph Orchestration** | Orchestrate client onboarding supervisors and sub-agents in LangGraph |
| 9 | **Monitoring & Security** | Implement model-drift alerts (Vertex) and K8s security best practices (OPA/Gatekeeper) |
| 10 | **Documentation & Best Practices** | Write coding standards, runbooks, and architecture diagrams for all above components |

— Each deliverable should include:

* **Code repo** with README + deployment scripts
* **Demo video** (5–10 min)
* **Architecture doc** (1–2 pages)

This plan ensures deep coverage of each technology, hands-on mastery, and clear individual responsibilities to accelerate your team’s ability to deliver an integrated Agentic-AI solution on GCP.