

# SRE Classroom Facilitator Guide

Thanks for signing up to facilitate a SRE classroom! Below are guidelines on how to make this experience awesome for you and your group.

## How does it work?

The SRE classroom includes a hands-on design exercise accompanied by topical presentations. Attendees are asked to design a reliable system for a specific scenario which usually mirrors an actual problem solved by Google in its production environment. The presentations are meant to teach relevant techniques to solve typical issues in distributed systems. Attendees are supposed to immediately apply these techniques.

## What do I need to do as a facilitator?

The facilitators help provide more focused guidance to individual groups. They help keep groups on track, making sure they continue making forward progress with the exercise.

- Be prepared: read the problem and sample solution ahead of time and make sure that you understand both. Provide guidance, without criticizing the solution.
- Enable your team without giving away solutions. You are not here to teach. Ask questions to guide the discussion: “Is specifying this detail important for the intent of the system”, or “How would you design this part of the system if you had a choice?”, or “Does this component satisfy the SLO?”
- Clarify what is important to the exercise and prevent following red herrings: asking questions is the right way to go. If you are stuck, escalate to the person running the workshop.
- Make sure that all group members are included in the discussion, e.g., ask for the opinion of other members if one person dominates the discussion.
  - Suggest a note-taker role, to ensure that a record of the discussion
  - “John’s had a lot of input; does anyone else have anything to add here?”
- Focus on the non-abstract part by asking questions that actually force the team to work out numbers when appropriate.

Because group experience may vary it is hard to come up with a rule on how much guidance is needed. However if a group defers decisions to the facilitator then it is most likely too much.

## Are there any milestones? Structure?

A good general approach is this:

1. Design a general architecture for a single data center first.
2. Extend the design to multiple data centers and tackle issues around being a distributed system.
3. Calculate actual hard resource requirements.

4. Identify potential reliability issues and correct numbers (e.g., SPOF, IOPS shortage).

**Is the sample solution the “correct” solution?**

Note that the sample solution is just that: a sample or an example. Encourage teams to come up with their own approach as long as it is technically sound.