Some amount of fine-grained deployment can be achieved by using existing capabilities. But to gain the most effective isolation between integrations implies the need to move to container-based technology. However, as we see, this is more than just a replatforming exercise. To gain the most significant benefits we need to move to a truly cloud-native style of deployment with impacts how teams build, deploy, administration, and monitor their integrations. Some enterprises want to take more gradual steps, staging their way to cloud-native, rather than jumping in with both feet.

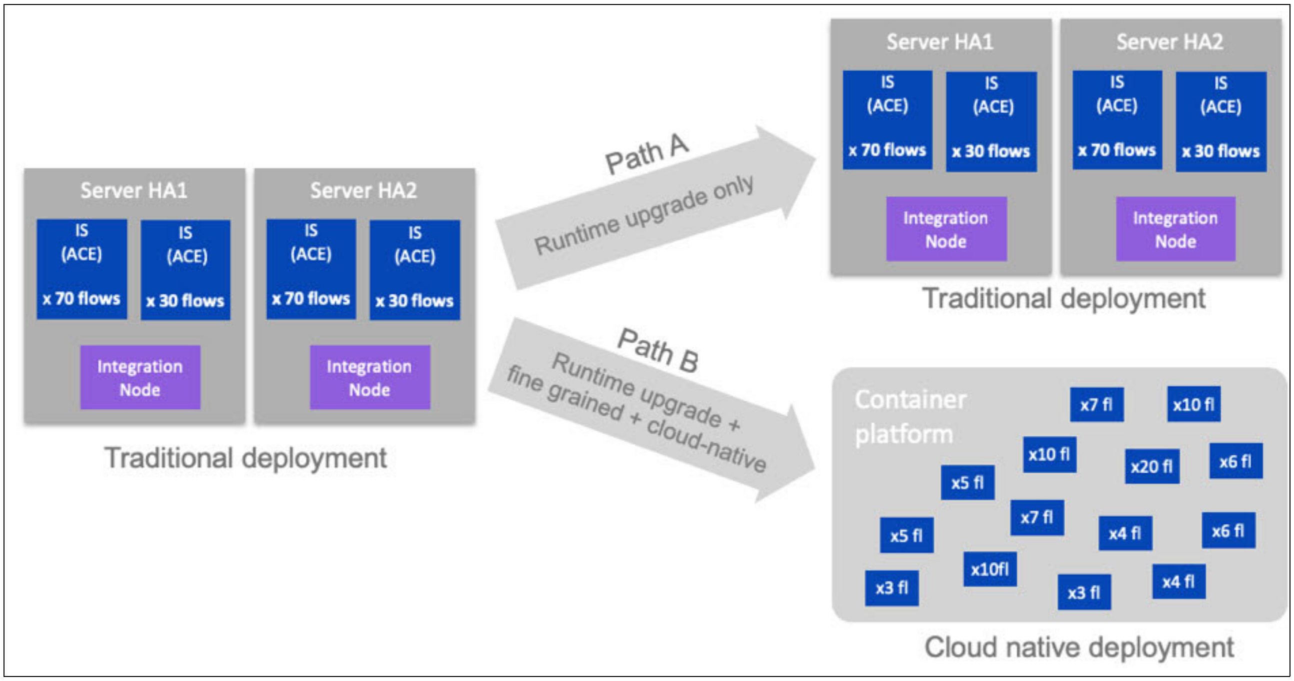

Figure 7-2 IBM App Connect adoption paths

We start with the conceptually simplest upgrade we could do, raising the level of the runtime to IBM App Connect v11, but still on our existing traditional topology (Path A). We then see how much we can push that toward the benefits of agile integration. And we can also see at what point we need to make the more significant changes toward a true cloud-native deployment on containers (Path B).

Path A: Runtime upgrade only (preserve existing topology)

On this path we simply upgrade the runtimes of the Integration Server and Integration Node, and the developer Toolkit. The core topology remains the same.

IBM App Connect v11 does not mandate a move to container infrastructure. It can still be deployed in this traditional topology by using Integration Nodes to administer the Integration Servers just as we did in prior versions of IBM Integration Bus. Let’s also assume that at least some of the workloads require a local MQ server, so that also needs to be present in the topology.

Even this simple upgrade path brings core runtime and tooling enhancements. Examples depend on what version you are moving from, but might include:

- Simpler file system-based installation
- Removed hard dependency on local MQ server
- New capabilities such as the Group node
- Toolkit now supported on MacOS