Cloud Run is a service for running containers and Google Cloud or in Kubernetes clusters.

Cloud Run is a compute service and it's primarily designed for stateless containers.

Now, stateless just means that if a container were to fail and start up again, that second instance of it doesn't depend on anything that was in memory in the first instance that died.

So basically, as long as you're not keeping state within your container, you will be able to use Cloud Run.

Now, Cloud Run is available in two forms, as a managed service in which case Google Cloud basically manages everything or by running it within Anthos. Anthos is a managed application platform that uses Kubernetes

CLoud run is basically pay per use. So a typical kind of Google model for pricing, there are limitations. with the managed service you can have up to a thousand services per region within a particular container. We are limited to two gigs of memory and two virtual CPUs.

Also, in terms of access control you can manage identities that can access the service.

You can also allow for unauthorized access. So if you wanna say, expose an application to the web and allow anybody to use it you know you can do that through unauthorized access. Also, you should know that cloud run executes in a G Visor sandbox and G Visor is a platform for basically giving containers each their own kernel.

It's a more secure model than other ways of running containers.

Now, Cloud Run has this concept of resources and there's three we wanna keep in mind. There's a service and that's the main abstraction. That's the thing that you are providing. A service may run in a single instance of a container or it may run in multiple instances, but each of those are not considered distinct services. Those are distinct instances, but providing one service.

Now, services are located in a region, but they are replicated across multiple zones for high availability.

Now, a revision is a deployment of a service and that consists of a specific image and a configuration. And then also there of course, are container instances and those are the things in which the revisions run.

And the container instance is auto scaled based on load by cloud run.

Now concurrences is an important topic when you're using something like cloud run or cloud functions. Now, in a cloud run by default container instances can receive up to 80 requests at the same time.

Now notice this is different from the way cloud function works which is more serial or, or sequenced where cloud functions can only have one request at a time.

Containers can handle multiple requests. There may be times where you don't want that

where you want to have basically just a, an instance focused or committed to just a single request at a time. And that can happen when each request consumes most

of the CPU and memory.

So if you have some kind of really compute intensive operation that you're doing and you don't wanna be thrashing between different requests, that's one reason to, to lower the concurrency to one. Or if you're just using an image that wasn't designed to handle multiple requests that can happen as well.

So those are a couple reasons where you want to think about concurrency with regards

to cloud run and adjust as needed.

Anthos is one of the most exciting services Google has announced in quite a while.

It's really a key to enabling hybrid and multi-cloud environments.

So Anthos is an application management platform and it's based on Kubernetes.

Now, basically the core thing that Anthos is designed to do is to be able to allow us as customers of Google Cloudand as cloud users to be able to easily move workloads

across different environments or different infrastructures.

For example, we have Anthos GKE, which runs in the Google Cloud and Google Cloud manages that. There's a version of Anthos for on-prem so you can run it in your own data center and there's also Anthos for AWS.

So we can have workloads running in Kubernetes that are managed by Anthos and we can move those workloads across these different infrastructures and that's really the big value add that Anthos brings.

Now, Anthos, as you might imagine is a fairly complex application platform.

at the top we have application development and then there's sort of application deployment mechanisms. And then really the key thing that I wanna talk about today are things like the policy management and service management, and then the lower levels are things like cluster management, like managing GKE clusters and infrastructure management like managing virtual machines. So Anthos has tools and layers of software for managing everything down to the virtual machine level at the infrastructure level, all the way through like policy management, configuration management,

up to being able to manage applications and deal with things like continuous integration and continuous deployment.

So Anthos supports microservices architecture which is no surprise since it's based on Kubernetes. It also comes with service mesh which includes Istio which is an open source project which is also used with Kubernetes and it does things like adds additional layers of services for us within the Kubernetes environment.

For example, it gives us fine grain control over how traffic gets routed. It does automatic monitoring so we can collect metrics and logs and traces without having to specify all the gory detail.

Also, it enables service-to-service authentication and authorization, which is really helpful, so you don't have to build it into each microservice.

Basically with Istio, the authentication or the Istio components run is something called a sidecar. Also, we have network access across the cloud and on premise, of course, because we're dealing in a hybrid and multi-cloud environment and we can do that using either VPNs which is probably the simplest way, or you could have a dedicated, or partner interconnect between the clouds, or clouds and on-prem. Now, another thing that Anthos does is it allows you to manage the configuration of your Kubernetes services in a centralized place.

So Kubernetes uses a declarative model for specifying configuration.

So for example, you can specify how many nodes you want running and when Kubernetes detects that your actual running environment is out of sync with your configuration, maybe you have fewer nodes than you specified in your configuration,

it'll automatically address that problem and add nodes as needed.

Now with Anthos, Anthos provides way of coordinating that configuration details, those configs, using GIT for version control.

Also, the config management service has a policy controller and that's helpful for enforcing business logic at the API request level. So basically, you can have rules applied when an API call is made by specifying them with this policy controller.

Another nice thing, as I mentioned, it automatically collects metrics, traces, logs and Anthos takes advantage of cloud logging

and cloud monitoring.

Now, cloud logging is used for basically storing logs, obviously, and the clusters automatically send log messages when dealing with messages about operations

and execution of things.

Now, there's also cloud audit log and that records messages about changes to sort of the control component.

So if someone's changing the configuration, or doing something related to your cluster,

that kind of thing is logged in the audit log.

Also, Kubernetes engine monitoring, the service, collects metrics for debugging and alerting and, again, this is done automatically, so you don't have to go around specifying

a lot of those details.