In order to minimize the time it takes to persist message data, Pub/Sub automatically stores published messages in the nearest Google Cloud region. Pub/Sub then delivers the messages to subscribers across the world, regardless of where the messages are stored.

In some cases, you may need more precise control of where your messages are stored. Pub/Sub's topic *message storage policy* offers a way to ensure that all data published to a topic is persisted in a specific region or set of regions, regardless of the publish request's origin. When multiple regions are allowed by the policy, Pub/Sub chooses the nearest allowed region.

Message storage policies can result in additional egress fees. See Resource location restriction implications
sub/pricing#storage-pricing) in the Pub/Sub pricing documentation.

A topic's message storage policy can be configured in the following ways:

- To configure all of the topics in an organization-wide scope, you can use the **Resource Location Restriction** organization policy. Organization-wide policies are maintained in the **Organization policies** section of the IAM & admin console (https://console.cloud.google.com/iam-admin/orgpolicies/list?project=_&service=pubsub.googleapis.com).
- For fine-grained control, you can configure a topic's message storage policy at topic creation, or via the `UpdateTopic` operation.

- If an explicit message storage policy **is not** specified when a topic is created, the new topic's message storage policy is automatically determined based on the effective **Resource Location Restriction** organization policy. When no organization policy is in effect, all regions are allowed.
- If an explicit message storage policy **is** in effect at topic creation time, it can allow only the regions specified by the effective **Resource Location Restriction** organization policy.

- When an organization policy is updated, the changes **do not automatically propagate** to existing topics. As such, an existing topic's message storage policy can get out of sync with the latest organization policy. See below for more information about resolving these differences if they occur.

- When a topic's message storage policy is updated, the changes **do not automatically propagate** to already-published messages. Messages already stored based on an older policy are **not** moved to be consistent with an the policy. Rather, the changes apply only to messages published after the update.

To configure message storage policy, you can either synchronize it with the organization policy or specify it explicitly for a topic. You can configure the policy using the:

- Topic details (https://console.cloud.google.com/cloudpubsub/topicList) view
- `gcloud` command-line tool
- Service API (using client libraries (/pubsub/docs/quickstart-client-libraries))

Update the existing policy on a topic with the current organization policy:

Set an explicit message storage policy on a topic, as a list of allowed Google Cloud regions:

This operation ensures that messages subsequently published to the topic are stored in `us-central1` or `us-east1`.

The Cloud Console displays any differences between the organization policy and individual topics' message storage policies. You can also synchronize a topic's message storage policy with the organization policy in Cloud Console. You cannot specify topic-level message storage policies in the Cloud Console.

To see which of your topics are out of sync with your organization policy, go to the console (https://console.cloud.google.com/cloudpubsub) and open the **Storage Policy** tab of the info panel on the right.

You can also examine the current policy using the command line.

To update one or more topics:

1. Go to the **Storage Policy** tab in the console (https://console.cloud.google.com/cloudpubsub).
2. Select one or more topics.
3. Click **Update**.

The policy specifies a list of allowed Google Cloud region names. As such, the following items are not supported:

- Exclusion lists
- Zones or multi-region locations

To help you understand where message data is stored, we offer Pub/Sub metrics (/monitoring/api/metrics_gcp#gcp-pubsub) broken down by Google Cloud region:

Counts of unacknowledged stored messages:

`subscription/num_unacked_messages_by_region`

Volume of stored data:

`subscription/unacked_bytes_by_region`

Age of oldest message:

`subscription/oldest_unacked_message_age_by_region`

Analogous metrics are available for topics and snapshots. In addition, corresponding metrics are available for acknowledged messages that are optionally retained for replay. For example:

`subscription/num_retained_acked_messages_by_region`

You can use these metrics to:

- Understand how your data is distributed across the world.
- Optimize publisher and subscriber deployment location, based on that data.

The message storage policy does not affect the overall SLA, but it does introduce an availability-control trade-off when publishers or subscribers run outside Google Cloud or in regions not allowed by the policy. Users who run publisher clients within the set of regions allowed by the message storage policy should not see any changes in the service's latency or availability.

To understand these trade-offs, it is worth considering how publish requests are routed. Generally, Pub/Sub attempts to store your messages as close as possible to the source of the request. Requests originating within Google Cloud are, as a rule, bound to the Pub/Sub instances in the same region. If a publisher is located in a single region, simply adding more regions to the message storage policy will not increase availability. When publishing from outside of GCP, an additional layer of routing is involved to get the request to a nearby GCP region where the Pub/Sub service is available.

Consider a message storage policy that allows only the `us-central1` region.

1. A publisher client running in `us-east1` issues a `Publish` request.
2. The request is routed to a Pub/Sub server in `us-east1`.
3. Rather than storing the data in `us-east1`, the request is routed to the nearest region allowed by the message storage policy, which is `us-central1`.
4. Pub/Sub stores the published messages in `us-central1` and forwards messages to subscribers from that location.

This mechanism has implications for request latency and overall system availability. Because the request traverses more network links, it takes longer to complete and has a relatively higher chance of failing.This also means that the subscribers might see the message somewhat later because it must travel to the nearest allowed region before being dispatched. If the policy allows a single region but your publisher applications run in multiple regions, the distributed application becomes only as available as the single allowed region.