|  |  |  |
| --- | --- | --- |
| 5. Cloud Functions | Working with Cloud Functions | Creating Cloud Function |
| Deploying Cloud Function via scripts |
| Trigger DAG |
| Unzip & Memory constraints |

Certainly! Here’s a detailed explanation of the topics you’ve inquired about regarding Google Cloud Functions:

* **Creating Cloud Function**:
  + **Purpose**: Write single-purpose functions that respond to cloud events.
  + **Process**: Choose a trigger, write the function code, deploy using the console or gcloud CLI, and set permissions.
  + **Languages**: Supports multiple languages like Python, Node.js, Go, etc.
* **Deploying Cloud Function via Scripts**:
  + **Automation**: Use scripts or CI/CD pipelines for deployment.
  + **Command Example**: ]
  + gcloud functions deploy FUNCTION\_NAME --runtime RUNTIME --trigger-http --entry-point ENTRY\_POINT --region REGION
* **Trigger DAG**:
  + **Integration**: Trigger Cloud Composer DAGs using Cloud Functions in response to events.
  + **Mechanism**: Use the Airflow REST API to trigger DAGs when events like file changes in Cloud Storage occur.
* **Unzip & Memory Constraints**:
  + **Handling Files**: Unzip files within Cloud Functions using libraries like zipfile in Python.
  + **Memory Limits**: Choose appropriate memory allocation based on the function’s needs, with limits up to 32GB for gen 2 Cloud Functions.

[For more comprehensive information, you can refer to the official Google Cloud documentation1](https://cloud.google.com/functions/docs/concepts/overview)[2](https://www.coursera.org/learn/developing-applications-with-cloud-functions-on-google-cloud)[3](https://codelabs.developers.google.com/codelabs/cloud-starting-cloudfunctions/). Feel free to ask if you need further assistance!

* Cloud Functions offers two product versions: Cloud Functions (1st gen), the original version, and Cloud Functions (2nd gen), a new version built on [Cloud Run](https://cloud.google.com/run) and [Eventarc](https://cloud.google.com/eventarc/docs) to provide an enhanced feature set. This page describes new features introduced in Cloud Functions (2nd gen) and provides a comparison between the two product versions.

Cloud Functions 2nd gen is a Function-as-a-Service product powered by Cloud Run and Eventarc. This enables users to leverage key benefits of Cloud Run including concurrency, traffic splitting, and longer processing time

Certainly! Let me explain the difference between **Container Registry** and **Artifact Registry**:

1. **Container Registry**:
   * **Purpose**: Container Registry is a service that allows you to store and manage container images.
   * **Contents**: It primarily stores **container images** (Docker images) used for deploying applications in containers.
   * **Features**:
     + Supports Docker images.
     + Provides multi-regional registry hosts.
     + Suitable for managing container images within the Google Cloud ecosystem.
   * **Availability**: Container Registry is still available and supported as a Google Enterprise API.
   * **Future Updates**: Going forward, new features will only be available in Artifact Registry.
   * [**Security**: Receives critical security fixes but won’t get major updates1](https://cloud.google.com/blog/products/application-development/understanding-artifact-registry-vs-container-registry).
2. **Artifact Registry**:
   * **Purpose**: Artifact Registry is a fully-managed service that extends beyond containers to support both **container images** and **non-container artifacts**.
   * **Supported Formats**:
     + Container images (like Container Registry).
     + Non-container artifacts, including OS packages (Debian, RPM) and language packages (Python, Java, Node.js).
   * **Unified Interface**: You can manage all these artifacts from a single, unified interface.
   * **Fine-Grained Permissions**: Artifact Registry offers fine-grained access control via Cloud IAM, allowing per-repository permissions.
   * **Regional Repositories**: You can create regional repositories directly in the location where they’ll be used for higher availability and speed.
   * **Cost Optimization**: Hosting repositories in the same region where they are used can result in cost savings due to free intra-region network traffic.

Top of Form

Bottom of Form

|  |  |
| --- | --- |
| **Cloud Functions (1st gen)** | **Cloud Functions (2nd gen)** |
| Image registry | Container Registry or Artifact Registry | Artifact Registry only |
| Request timeout | Up to 9 minutes | * Up to 60 minutes for HTTP-triggered functions * Up to 9 minutes for event-triggered functions |
| Instance size | Up to 8GB RAM with 2 vCPU | Up to 16GiB RAM with 4 vCPU |
| Concurrency | 1 concurrent request per function instance | Up to 1000 concurrent requests per function instance |
| Traffic splitting | Not supported | Supported |
| Event types | [Direct support for events from 7 sources](https://cloud.google.com/functions/docs/calling#1st-gen-triggers) | [Support for any event type supported by Eventarc](https://cloud.google.com/functions/docs/calling#2nd-gen-triggers), including 90+ event sources via Cloud Audit Logs |
| CloudEvents | Supported only in Ruby, .NET, and PHP runtimes | Supported in all language runtimes |
| Security Updates | [When function is deployed or redeployed](https://cloud.google.com/functions/docs/securing/execution-environment-security#security_update_policy) [Automatic updates](https://cloud.google.com/functions/docs/securing/execution-environment-security#security_update_policy) | [When function is deployed or redeployed](https://cloud.google.com/functions/docs/securing/execution-environment-security#security_update_policy) |

Certainly! Let’s explore how you can use **Cloud Functions** to respond to changes in a **Cloud Storage bucket**, including archiving files. When you specify a **Cloud Storage trigger** for a function, it will be called whenever specific events occur within the specified bucket. Here are the key details:

1. **Event Types Supported**:
   * **Object Finalized**:
     + Event Type: google.cloud.storage.object.v1.finalized (2nd gen) or google.storage.object.finalize (1st gen).
     + Occurs when a new object is created or an existing object is overwritten, resulting in a new generation of that object.
   * **Object Deleted**:
     + Event Type: google.cloud.storage.object.v1.deleted (2nd gen) or google.storage.object.delete (1st gen).
     + Triggered when an object is permanently deleted.
   * **Object Archived**:
     + Event Type: google.cloud.storage.object.v1.archived (2nd gen) or google.storage.object.archive (1st gen).
     + Occurs when a live version of an object becomes a noncurrent version (see Object versioning for details).
   * **Object Metadata Updated**:
     + Event Type: google.cloud.storage.object.v1.metadataUpdated (2nd gen) or google.storage.object.metadataUpdate (1st gen).
     + Triggered when the metadata of an existing object changes.
2. **Implementation as an Event-Driven Function**:
   * Use either of the following approaches:
     + **CloudEvent Function**:
       - Cloud Storage event data is passed to your function in the CloudEvents format.
       - The CloudEvent data payload is of type StorageObjectData.
     + **Background Function**:
       - The Cloud Storage event data payload is passed directly to your function in the StorageObjectData format.
3. **Permissions**:
   * For Cloud Functions (2nd gen), the Cloud Storage service agent must have the **Pub/Sub Publisher** (roles/pubsub.publisher) IAM role on your project.
   * This requirement does not apply to Cloud Functions (1st gen).
4. **Deployment**:
   * Specify a Cloud Storage trigger when you deploy a function.
   * If using the gcloud CLI, you can use the **Cloud Storage Object Finalized** event type with appropriate flags.

Remember that the function and the bucket must be in the same Google Cloud project for the trigger to work.