Events: An event is the smallest amount of information that fully describes something that happened in the system.

* Discrete events: state change, actionable (invoice has been created)
* Series events: report a condition, analyzable (stream of events belonging together, eg. Telemetry readings, loggings)

# Azure Event Grid

Use this, when my app deals with discrete events, deals with publisher-subscriber model and to handle event, but not the data.

An event of size up to 64 KB is covered by General Availability (GA) Service Level Agreement (SLA). The support for an event of size up to 1 MB is currently in preview. Events over 64 KB are charged in 64-KB increments.

Topic: an endpoint where the source can send events

* Publisher creates the event grid topic, decides whether an event source needs 1 or more topics
* Used for a collection of related events
* Subscribes decide which topics to subscribe to
* System topics: by Azure resources (invisible, owned by publisher owns, but can be subscribed to)
* Custom topics: third party topics (I can create, etc)
* Orders Topics, Users Topics, etc (for small apps, 1 is enough)

Event subscriptions: tells Event Grid which events on a topic we’re interested in receiving

* And endpoint is create, in which filtering by event type, or subject pattern is possible
* Expiration: can be set (eg. Testing purposes, no need for clean up is needed then)

Event handlers: A place where the event is sent, then the handler takes further actions to process the event

* Azure services, webhooks
* For HTTP webhooks, Event Grid waits for a 200 – OK

Batching: when using custom topic, events must be published in an array: up to 1MB

Needs a provider:  
az provider register --namespace Microsoft.EventGrid

Create topic:  
topicname=<your-topic-name>

az eventgrid topic create --name $topicname -l westus2 -g gridResourceGroup

Subscribe to topic:

endpoint=https://$sitename.azurewebsites.net/api/updates

az eventgrid event-subscription create \

--source-resource-id "/subscriptions/{subscription-id}/resourceGroups/{resource-group}/providers/Microsoft.EventGrid/topics/$topicname"

--name demoViewerSub

--endpoint $endpoint

At this point, Event Grid send a validation event: Event Grid sends the validation event so the endpoint can verify that it wants to receive event data.

Trigger an event:

1. Get URL & key:

endpoint=$(az eventgrid topic show --name $topicname -g gridResourceGroup --query "endpoint" --output tsv)  
  
key=$(az eventgrid topic key list --name $topicname -g gridResourceGroup --query "key1" --output tsv)

1. Post some JSON to it  
   event='[ {"id": "'"$RANDOM"'", "eventType": "recordInserted", "subject": "myapp/vehicles/motorcycles", "eventTime": "'`date +%Y-%m-%dT%H:%M:%S%z`'", "data":{ "make": "Ducati", "model": "Monster"},"dataVersion": "1.0"} ]'