# AZ300-something Notes

Export Azure diagnostic log data to:

* Azure Storage
* Event Hub: subscribe to event data funneled into the hub
* Azure Monitor logs: Kusto Query Language (KQL) to query

**Logs** are **blob** objects,  
**Metrics** (measured values, counts representing time series 🡪 they form the baseline) as **tables**.

Configure log collection for public IP address resource:  
- Monitor  
- Settings, Diagnostic settings  
- Add diagnostic setting: logs/metrics are resource specific (eg DDoS logs for IP)  
(we could have done this via Resource Manager API, Azure CLI or PowerShell)

**Alerts:**  
- Azure Monitor  
- Alerts  
 New Alert Rule (we can select from multiple Azure subscriptions)  
 - **Trigger:** can be either **Metrics** or **Activity Log** (listen to particular events)  
 Actions, Add: send email, sms, or trigger Function or, Logic App, send webhook, start Azure  
 Azure Automation runbook or integrate with our IT Service Management platform

We can even create a KQL query and use it in an Alert as a rule

**Action groups:** take action automatically when an alert is fired  
Generally, best practice is to create Action Groups first, then reuse them later on in alert definitions  
- Azure Monitor  
- Manage Actions, Add Action Group (specify subscription and resource group)

**Log Analytics:** collect Activity logs from multiple subscriptions (KQL)

Unused Resources: big cost saving opportunity  
**Azure Advisor** -> Cost recommendations

Price (historic data) 🡪 Cost Management + Billing  
Price (forecast) 🡪 Azure Monitor + Usage and Estimated Costs

**Storage account:** name must be globally unique (each service is bound default to public Azure Resource Manager API endpopint like http://account-name.blob.core.windows.net

* General purpose v2 (blob, table [NoSql], queue, file)
* General purpose v1 (backwards compatibility only, has no tiers nor zones)
* Blob storage (backwards, only stores VM virtual hard disks – nowadays put them in managed disk storage instead)

Hot tier: discount on transaction cost (data frequently accessed)  
Cool tier: discount on data storage (data not frequently accessed)  
Archive: only access data on special occasions (we must rehydrate blobs first which have cost)

**Locally Redundant Storage (LRS):** 3 copies of Storage Account, in single datacenter  
**Zone Red. Stor.:** 3 copies, same region, but different datacenters  
**Geo Red. Stor.:** 3 copies in home region, 3 copies in second (paired) region [these are close to each other]  
**Read access geo red. Stor.:** Same as geo redundant, but we can access the 2ndary region, it has a unique URL <account-name>.secondary<service>.windows.net

**Access keys:** 2, 512 bit interchangeable access key for Storage Accounts (can be in Key Vault)  
SAS: **Shared Access Key**:  
- Protocol (HTTPS)  
- Address (fully qualified path of storage acc)  
- Permissions (CRUD)  
- Time interval