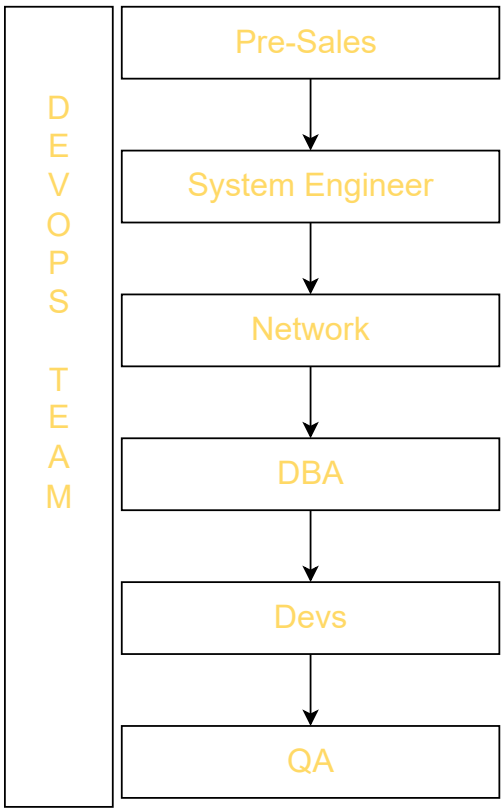
What is Devops?

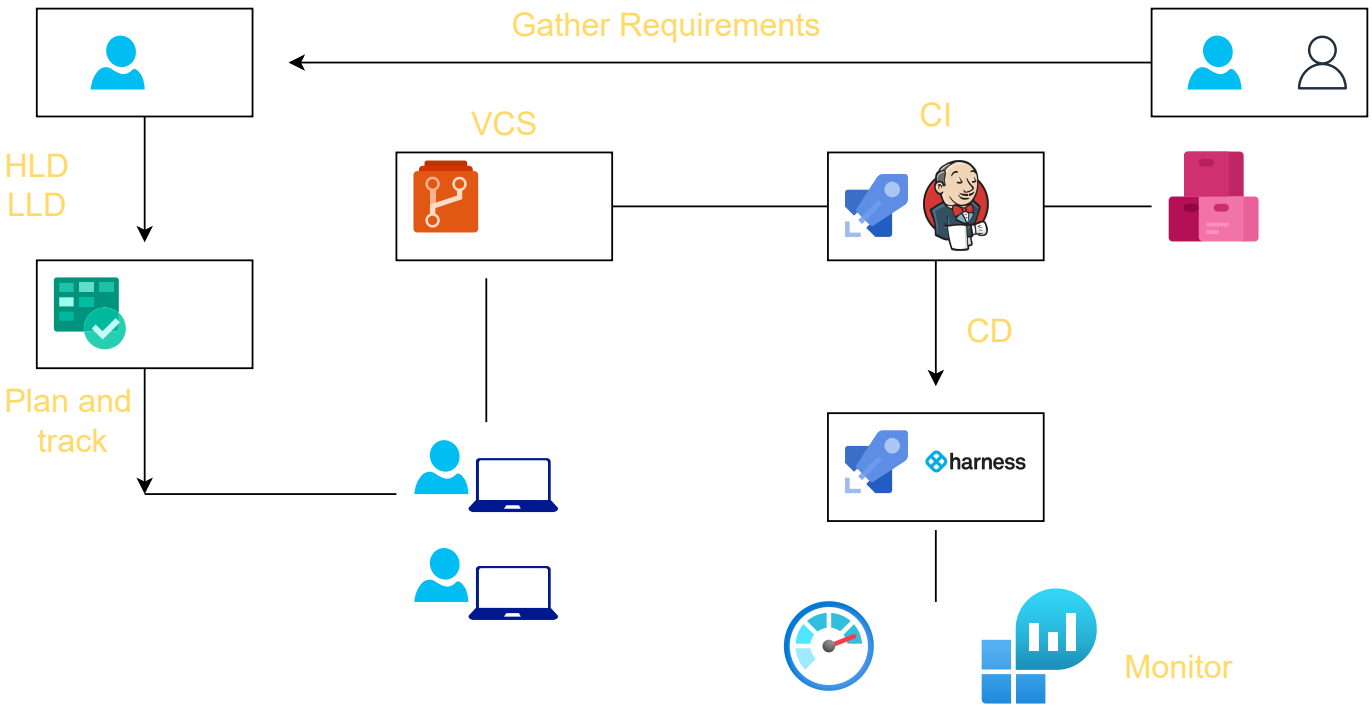
1. faster releases
2. Quality and efficiently
3. Continuous Improvement

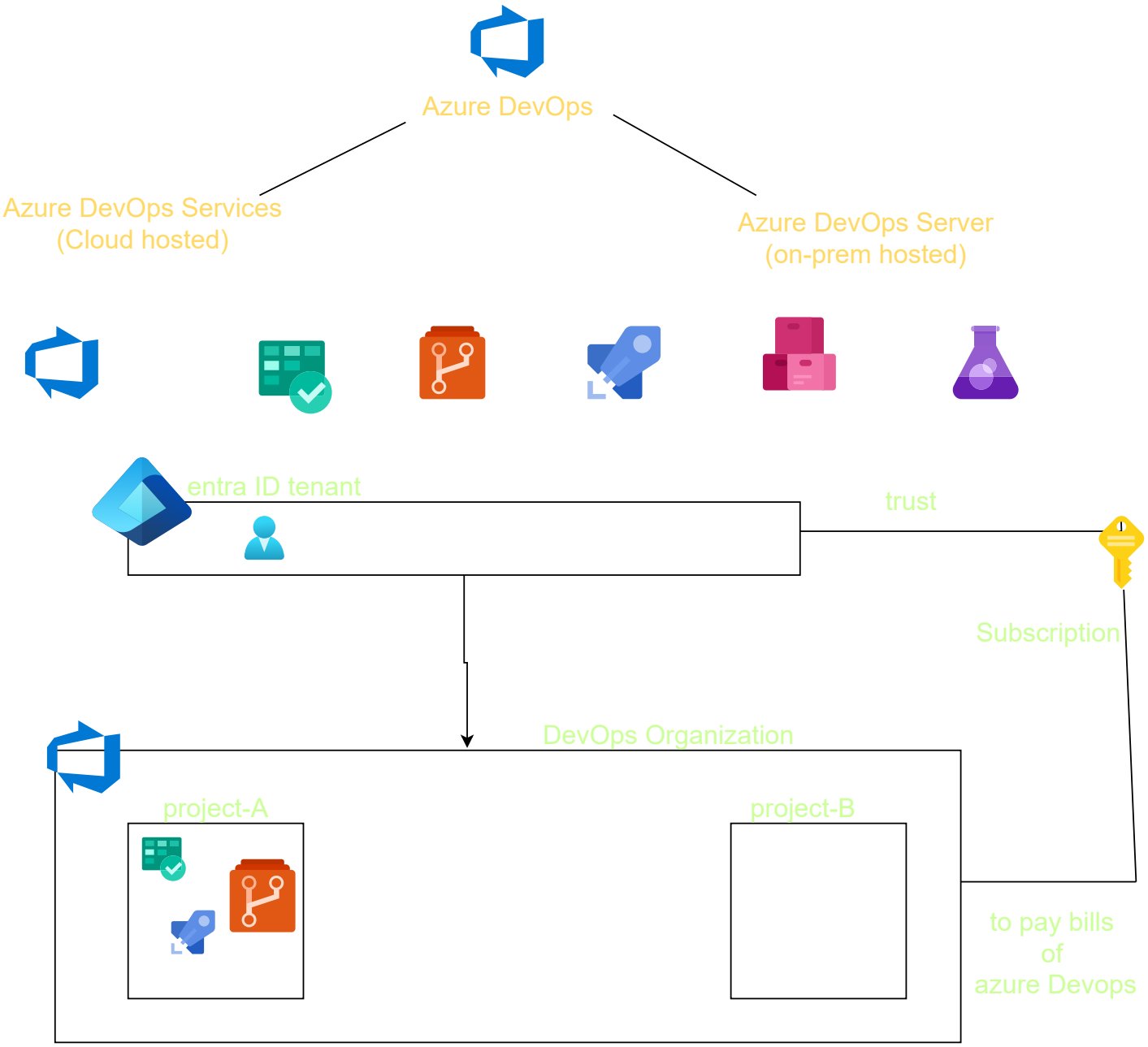
People, Process and Products

We cannot buy or Sell DevOps
build it ....



SDLC (software Dev Lifecycle):





Access Level in DevOps:
What can you see

All users in Azure DevOps belong to one or more default security groups.
Security groups get assigned
permissions that either Allow or Deny access to features or tasks.

Product management is an organizational function that guides every step of a product's lifecycle — from development to positioning and pricing — by focusing on the product and its customers first and foremost.

Running the scrum calls, creating those user stories ==> Scrum Masters

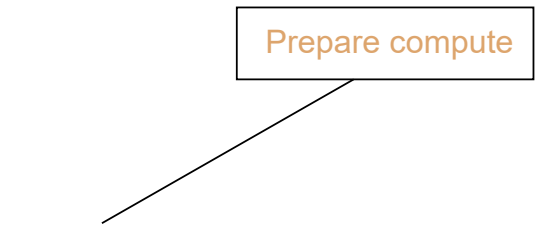
- Use **Agile** when you need flexibility and can afford to adapt to change constantly.
- Use **Scrum** when you need a more structured, time-boxed approach with specific roles and regular feedback loops.

Anything you create inside azuree boards ==> work item
example: task, epic, bug etc

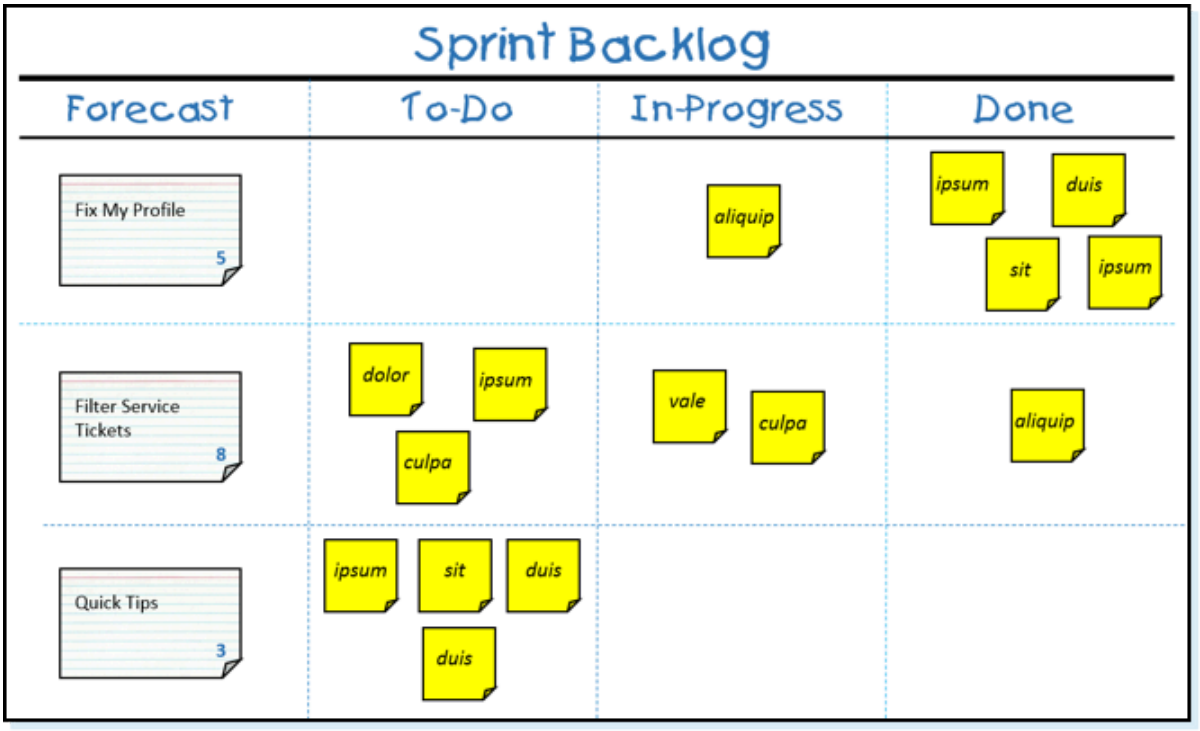
wrong work item process type. No problem. You can change it after creation.

Select Work item Process type
a, change it after creation (possible)

Game Development:
1. Game Character
2. Game Story
3. Create Animation
4. prepare compute
5. Prepare Networking database
6.



What is a product backlog? A product backlog is a prioritized list of work for the development team that is derived from the product roadmap and its requirements. The most important items are shown at the top of the product backlog so the team knows what to deliver first

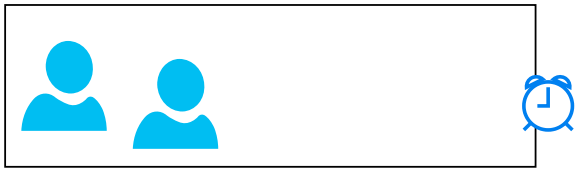


A kanban board is an agile project management tool designed to help visualize work, limit work-in-progress, and maximize efficiency (or flow). It can help both agile and DevOps teams establish order in their daily work.

Collection of work item is called as Area

Task if you have to perform in your entire graduation
Product backlogs

Team - group of people having common work items



sprint backlogs

sprint

Queries - filter work items

A burndown chart is a graph that represents the work left to do versus the time it takes to complete it. It can be especially useful for teams working in sprints, as it can effectively show whether your deadlines are able to be met along the way.

https://learn.microsoft.com/en-us/azure/devops/boards/work-items/guidance/media/alm_pt_wits_testexperience.png?view=azure-devops

Frontend - ReactJS, ExpressJS, HTML, CSS, Kotlin, Swift

Backend - Python, Java, Node

Database: MongoDB, MySQL

Terraform, Ansible, Bash

Dockerfiles, K8s manifests

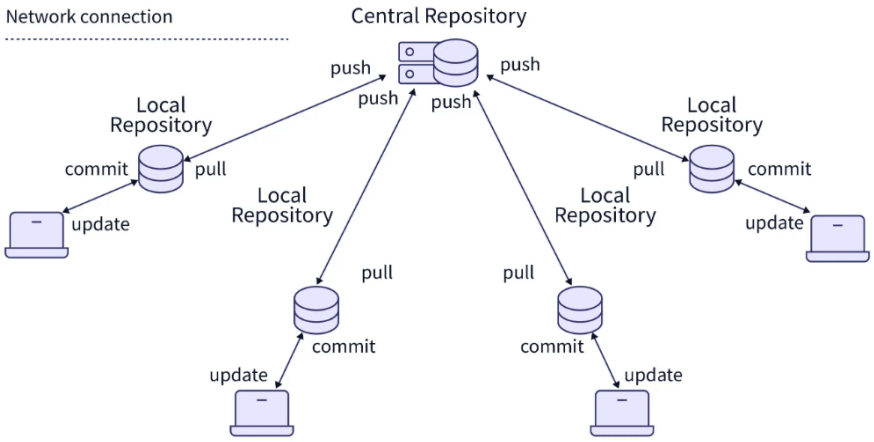
1. Central Place
2. avoid conflicts
3. version control

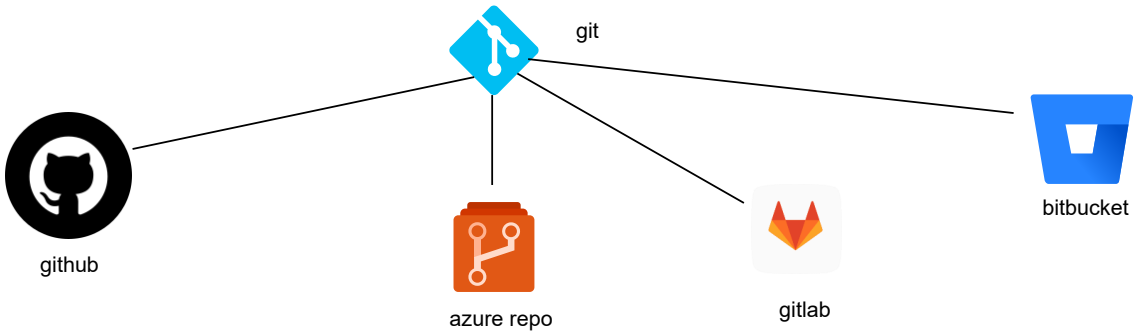
→ Version control system (VCS)
aka
SCM (source code management)

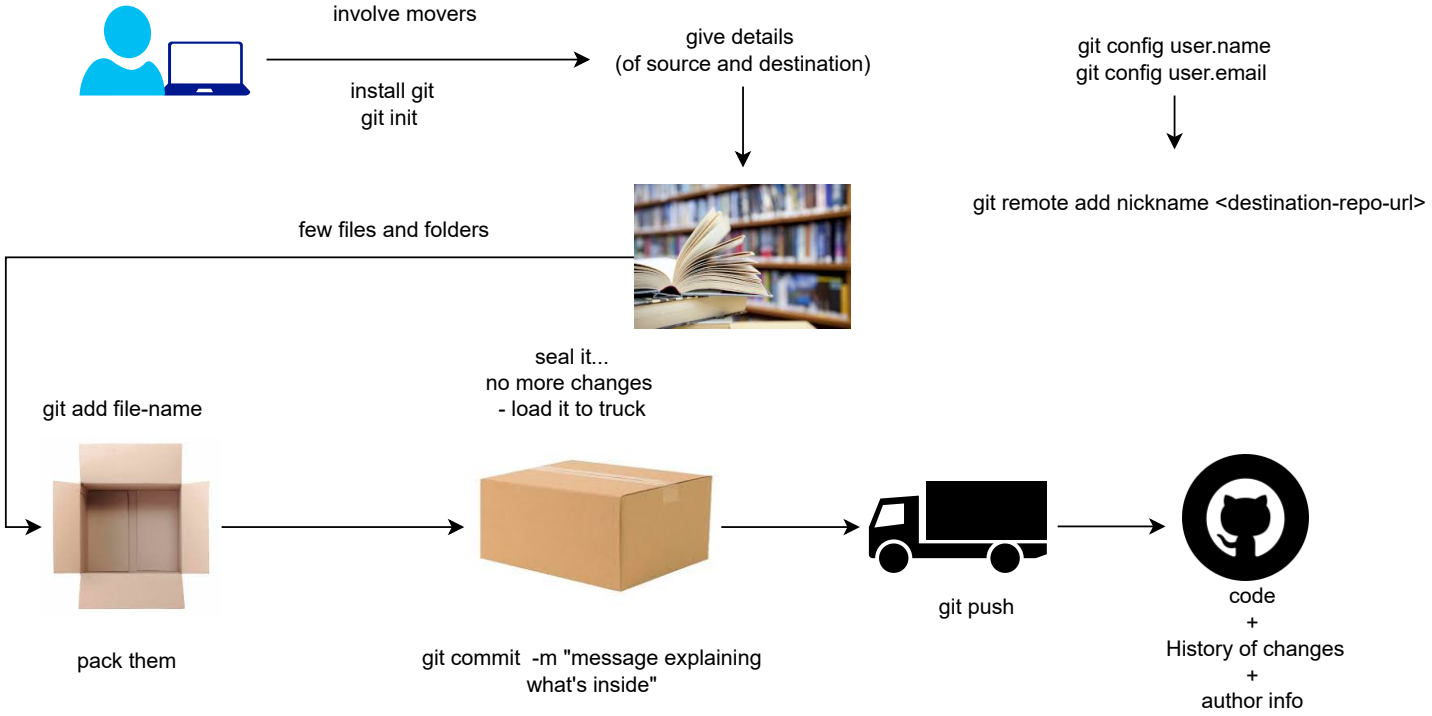
↓

git : distributed VCS

A distributed version control system (DVCS) brings a local copy of the complete repository to every team member's computer, so they can commit, branch, and merge locally. The server doesn't have to store a physical file for each branch — it just needs the differences between each commit.

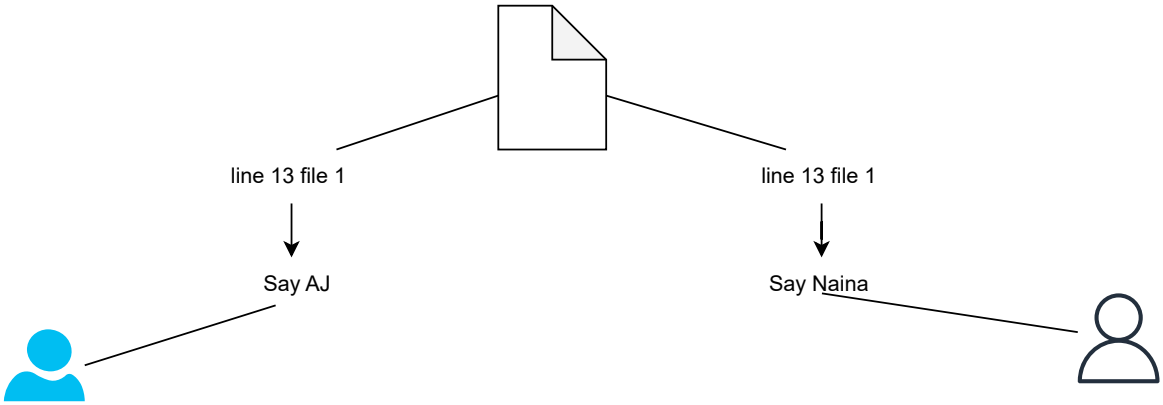

Network connection
Central Repository
push
push
push
push
Local Repository
Local Repository
Local Repository
Local Repository
commit
pull
pull
commit
update
update
pull
pull
commit
commit
update
update

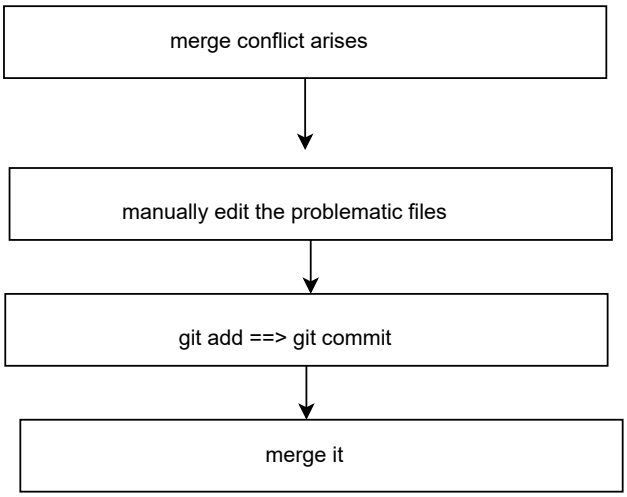

git
github
azure repo
gitlab
bitbucket


involve movers
install git
git init
give details
(of source and destination)
git config user.name
git config user.email
git remote add nickname <destination-repo-url>
few files and folders
seal it...
no more changes
- load it to truck
git add file-name
pack them
git commit -m "message explaining what's inside"
git push
code
+
History of changes
+
author info

Trying to combine (merge) code of two different branches

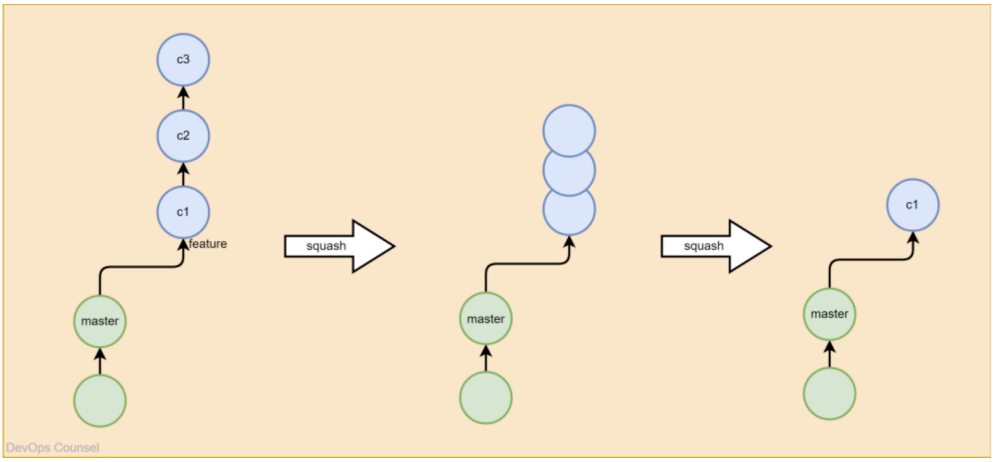
two different people try to make change on the same line of the same file. ==> Merge Conflicts



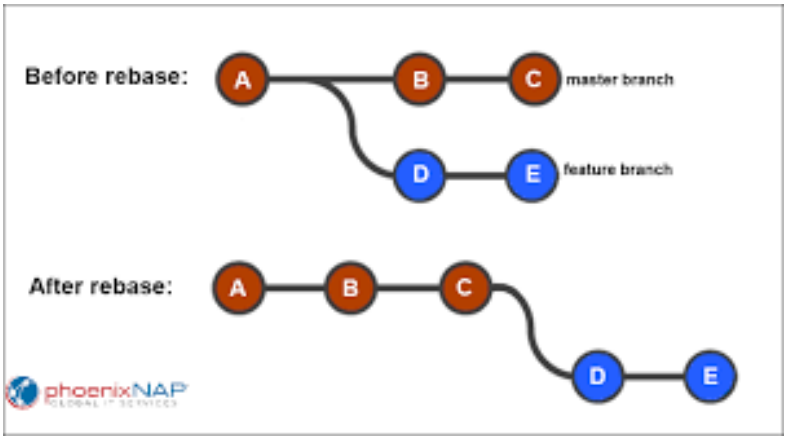



```
<<<<<<< chnage2-user2
   bucket = "AJ-the-user-tfstate"
   key    = "test/test.tfstate"
   region = "us-east-1"
   # use_lockfile = true
   dynamodb_table = "mumbai-dynamodb"
=======
   bucket = "tokyo-dev-env-tfstate"
   key    = "test/tokyo.tfstate"
   region = "ap-northeast-1"
   use_lockfile = true
   dynamodb_table = "tokyo-dynamodb-table"
>>>>>>> chnage1-user1
```

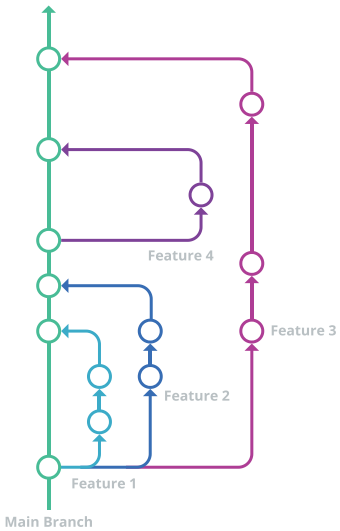
git squash

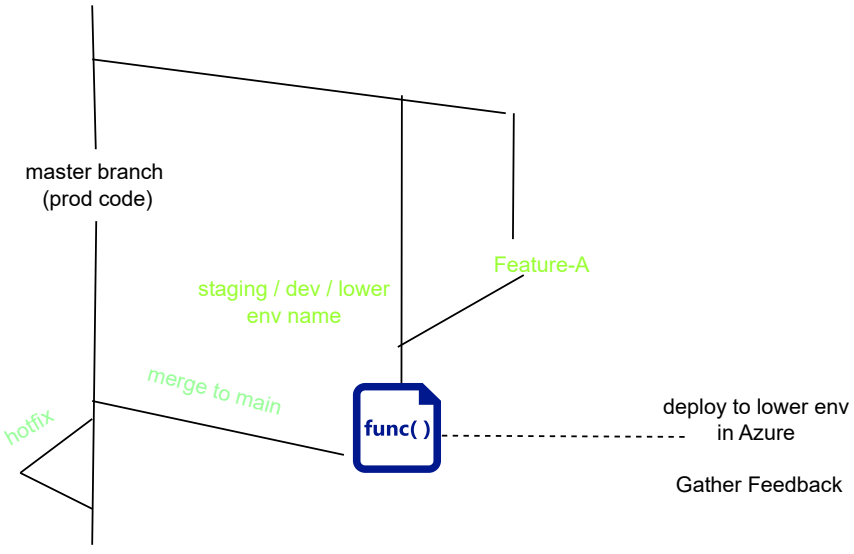


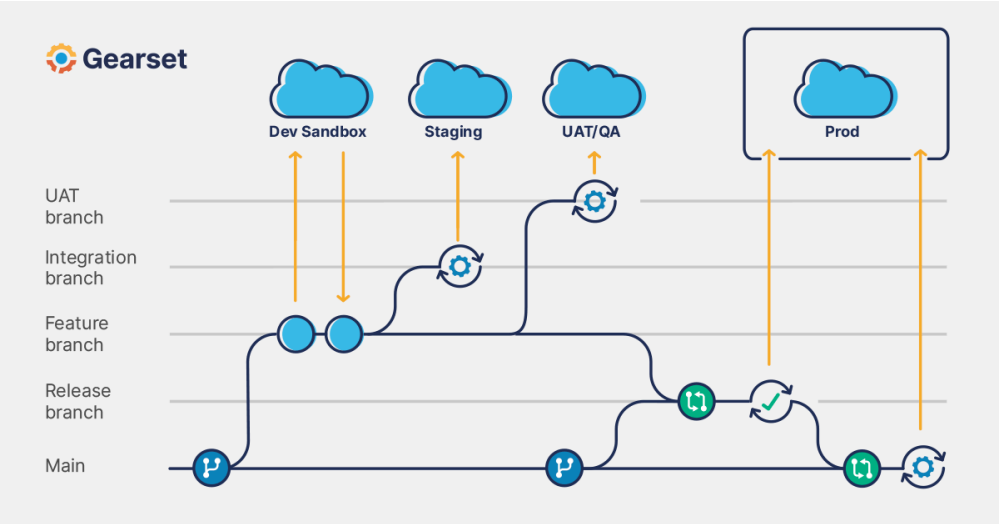
git rebase



git merge commit







Pipeline

Manual - time and efforts

When you code :
1. No more manual efforts
2. Idempotence : Produce same result every time
3. Easy to share and collaborate

→ DO NOT REPEAT YOURSELF so Create A TEMPLATE

↓

specific event occurs ==> run this template

translator in between, correct? Who translates my English into the binary for computers, you understand?

↓

build Process (Java, CS, Golang)

↓

exe

build ==> test ==> Produce Output (aka Artifacts)

Build pipeline

deploy it to non prod ==> test ==> get approval

↓

Deploy to Prod

Deploy to Env

CPU, Memory etc to RUN pipeline tasks → Pipeline agent → Collection of such agents => agent Pool

```
##[error]No hosted parallelism has been purchased or granted. To request a free parallelism grant, please fill out the following form https://aka.ms/azpipelines-parallelism-request
```

Attach billing and purchase paid parallel job fir pipeline

Define a Pipeline

Trigger: When to Run
a. Code change
b. PR
c. schedule
d. link to other pipeline

define tasks: what to do

Job: collection of task

every pipeline requires atleast one job

Agent:

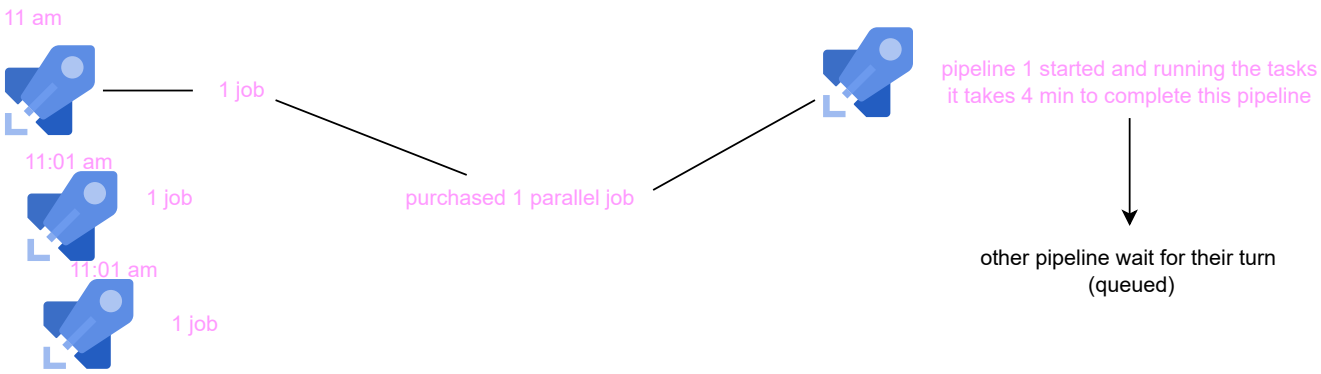
Microsoft-hosted agents

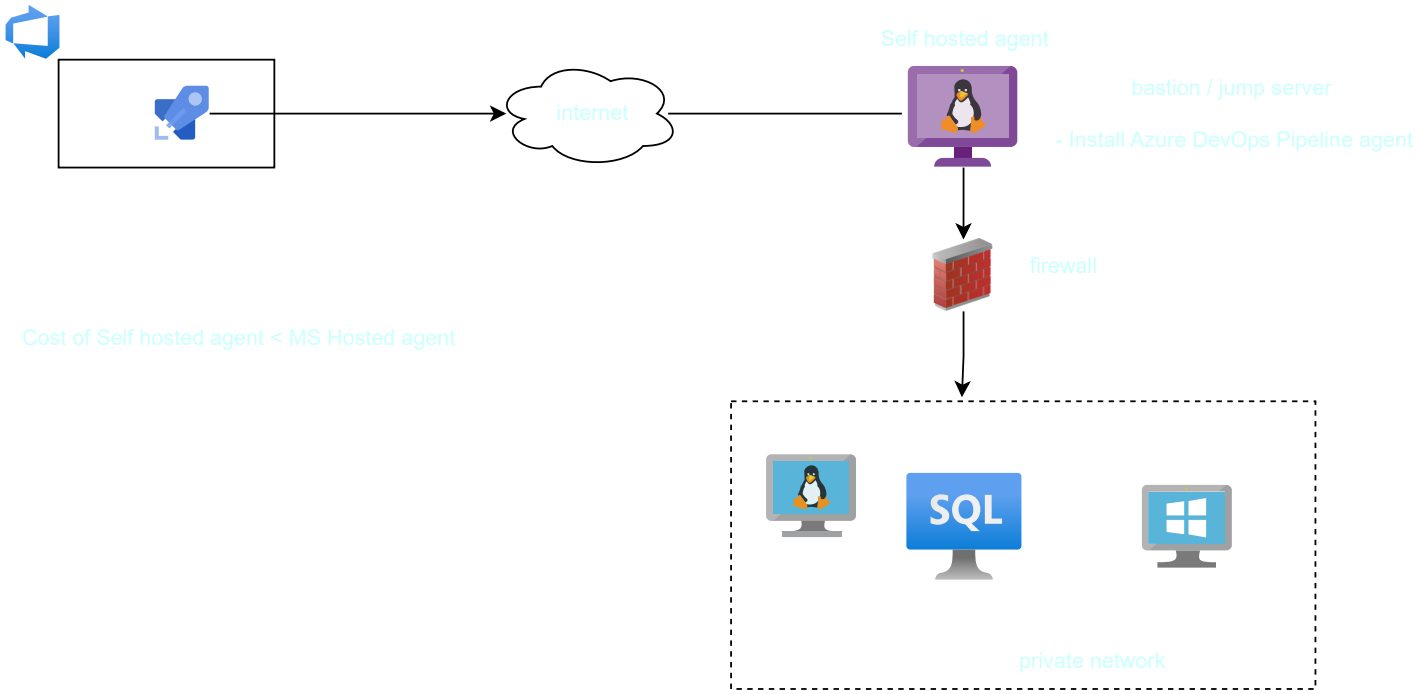
If your pipelines are in Azure Pipelines, then you've got a convenient option to run your jobs using a **Microsoft-hosted agent**. With Microsoft-hosted agents, maintenance and upgrades are taken care of for you. You always get the latest version of the VM image you specify in your pipeline. Each time you run a pipeline, you get a fresh virtual machine for each job in the pipeline. The virtual machine is discarded after one job (which means any change that a job makes to the virtual machine file system, such as checking out code, will be unavailable to the next job). Microsoft-hosted agents can run jobs directly on the VM or in a container.

Self-hosted agents

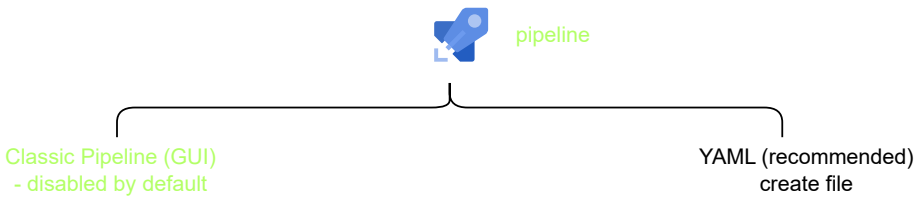
An agent that you set up and manage on your own to run jobs is a **self-hosted agent**. You can use self-hosted agents in Azure Pipelines or Azure DevOps Server. Self-hosted agents give you more control to install dependent software needed for your builds and deployments. Also, machine-level caches and configuration persist from run to run, which can boost speed.

- $40 per extra Microsoft-hosted CI/CD parallel job and $15 per extra self-hosted CI/CD parallel job with unlimited minutes





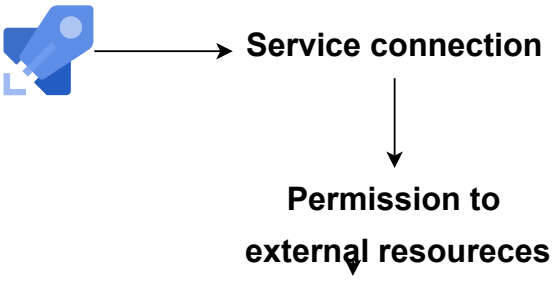
Managed DevOps Pools empowers development teams to quickly and easily spin up Azure DevOps agent pools that are tailored to a team's specific needs. Managed DevOps Pools implements security best practices, provides knobs to balance cost and performance, provides paths for the most common scenarios, and significantly reduces time spent in creating and maintaining custom pools.



Off

Disable creation of classic build pipelines

No classic build pipelines can be created / imported. Existing ones will continue to work.



Off

Disable creation of classic release pipelines

No classic release pipelines, task groups, and deployment groups can be created / imported. Existing ones will continue to work.

DevOps Project

External Tool / Service

forwards request to

Toolchain

Service connetcion

Entra ID

Service Principal / App Registrations / Managed Identity

RBAC (Role based access control)

- can be attached at multiple levels

- fine grained permissions

In Cicd should be involved a human being for approvals ?

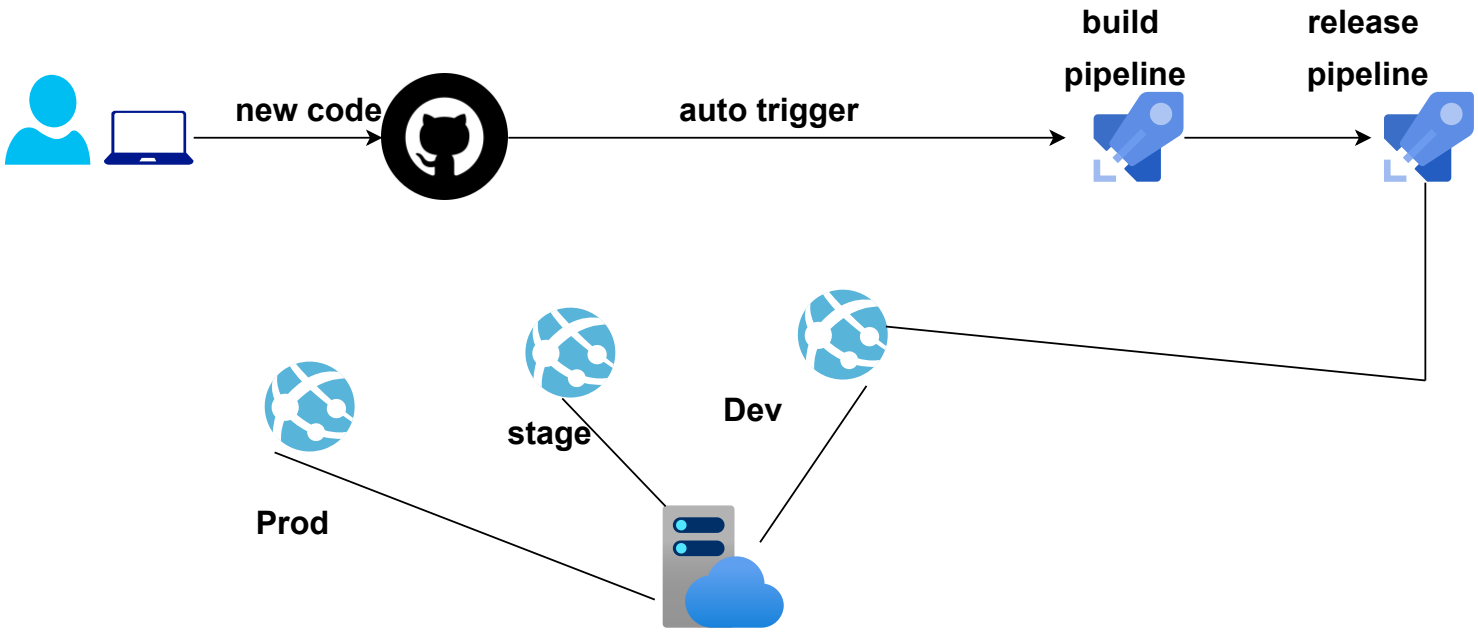
classic Editor (Build Pipeline) → classic Release → Deploy

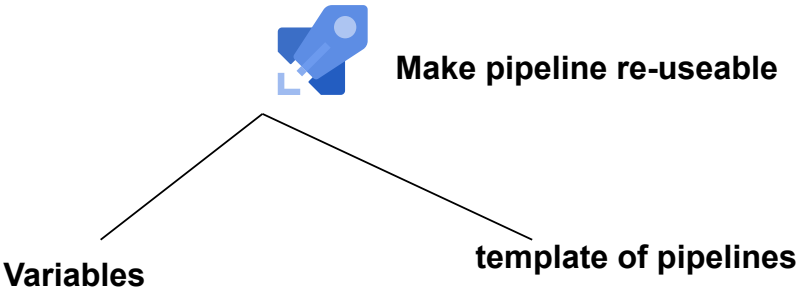
Approvals

Release Gates :

conditions that you need to check b4 deployments

→ variables in pipeline





Build.Reason

Build.BuildId

Pre-defined vars

network infrastructure related files.

working directory:
pipeline variable

network infrastructure related files.

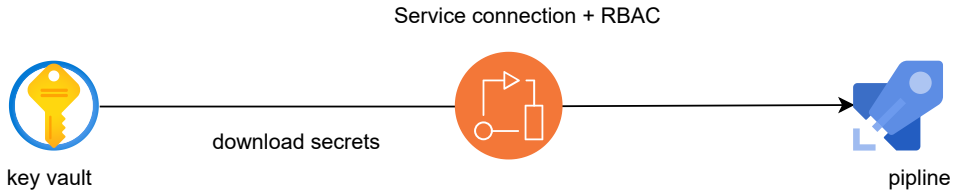
Subscription_id: fixed variable for every pipeline part of same project

list of predefined variables that are available for your use.

hese variables are automatically set by the system and read-only

Ananay Ojha
They are case sensitive.

Azure Key Vault is a cloud service for securely storing and accessing secrets. A secret is anything that you want to tightly control access to, such as API keys, passwords, certificates, or cryptographic keys. Key Vault service supports two types of containers: vaults and managed hardware security module(HSM) pools.



Creating the secret 'test'.

An error occurred while creating the secret 'test'. → RBAC

OR use 1 common agent for entire pipeline →

job: 1 ==> MS Agent
using terraform create infra

init.... plan....validate.....terratest.....apply

job 2 ==> self hosted agent
fetch db creds and now put them in configurationfile from hashicorp vault

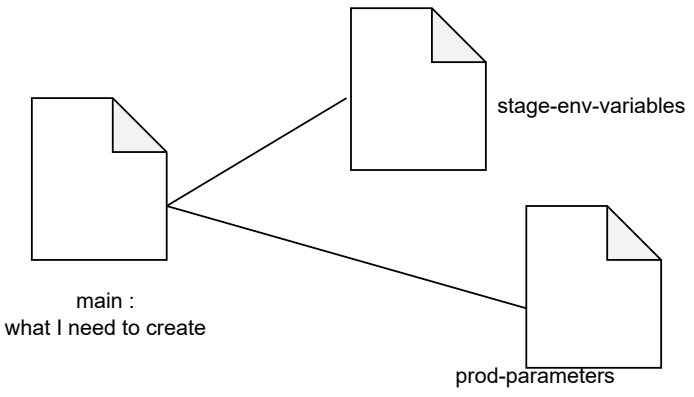
job 3 ==> docker container as agent
build and publish golang / python / java code

Example-2:

1. Job that deploys to Cloud: Cloud hosted agents
2. Job2: that deploys to on-prem : on-prem agent
job3: runs a PowerShell script => use windows MS hosted agent

Templates let you define reusable content, logic, and parameters in YAML pipelines. To work with templates effectively, you need to have a basic understanding of Azure Pipelines key concepts such as stages, steps, and jobs.

terraform : IaC = ARM template = Bicep = CFT (AWS) →



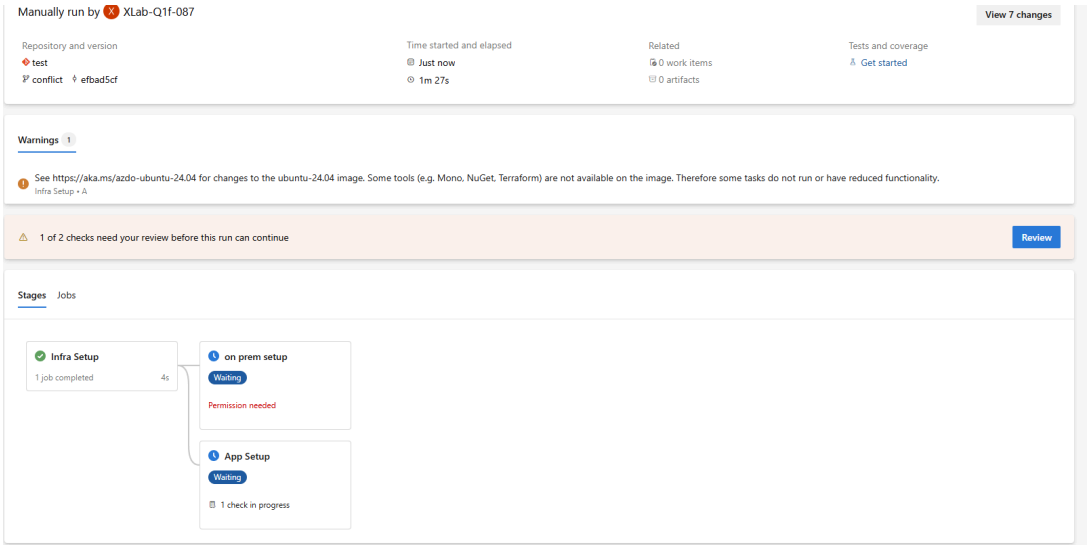
Infrastructure as code (IaC) is the ability to provision and support your computing infrastructure using code instead of manual processes and settings. Any application environment requires many infrastructure components like operating systems, database connections, and storage.

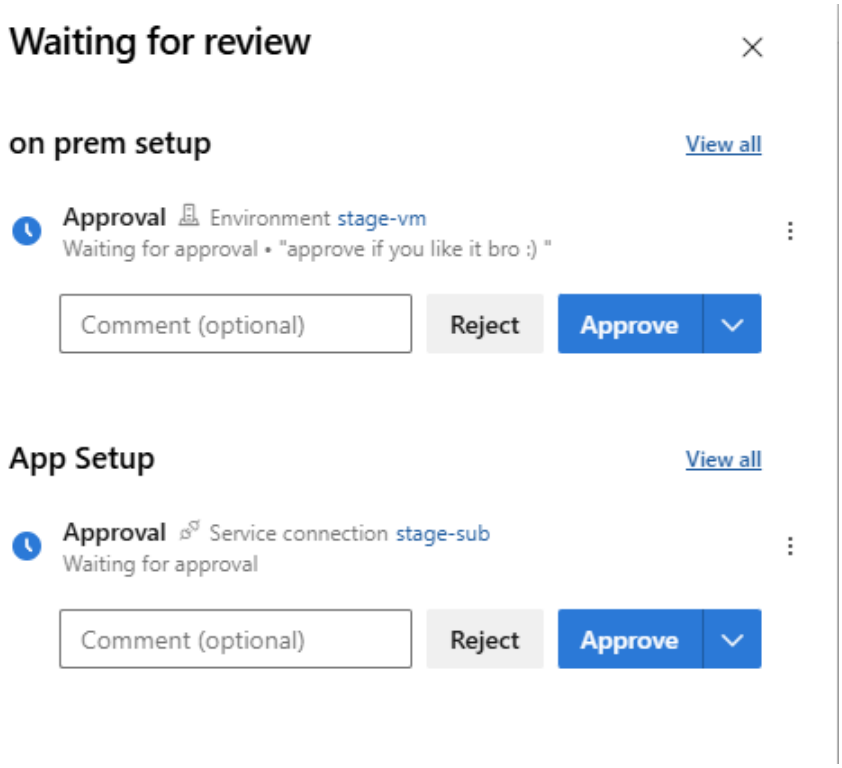
Approvals IN YAML :

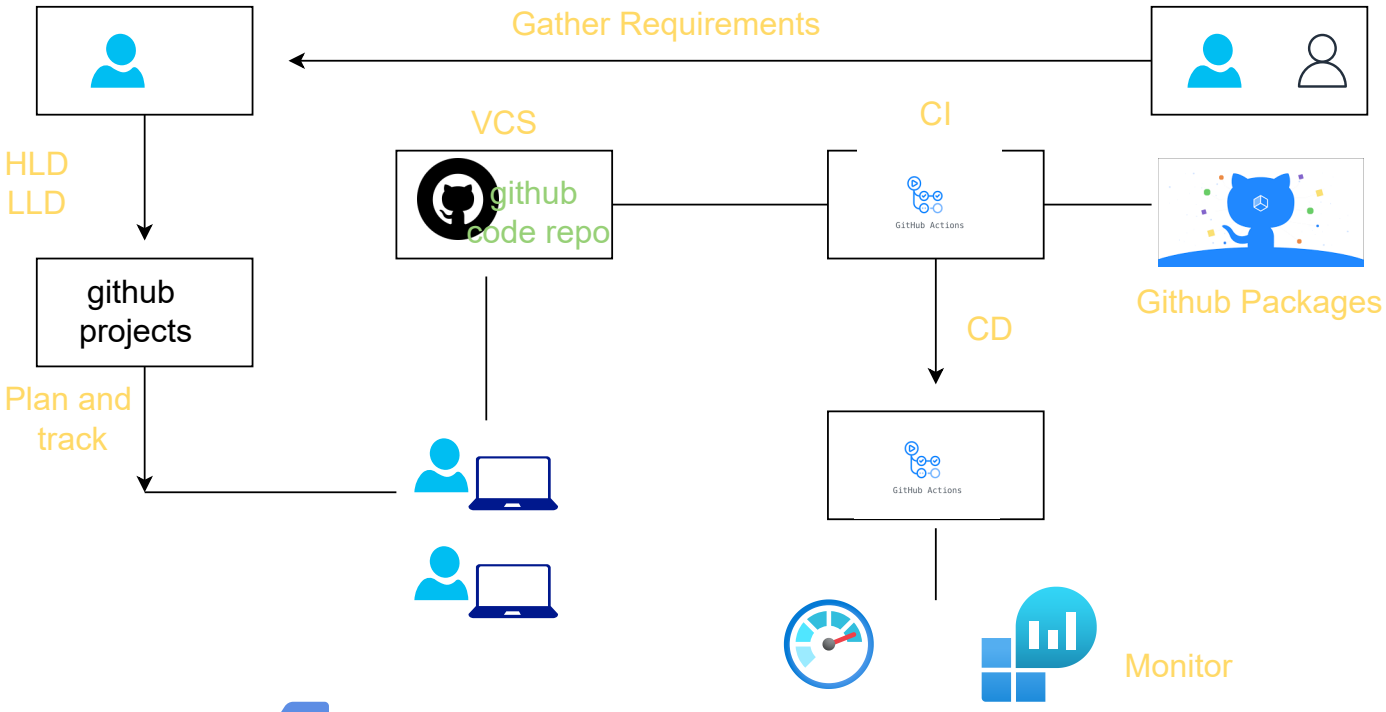
1. Service connection:

2. Environments : (Any on-prem VM or K8s Cluster thats connected to Azure DevOps via Internet)

An environment represents a logical target where your pipeline deploys software. Typical environment names are Dev, Test, QA, Staging, and Production.







trigger	trigger	on
jobs	tasks	job
agents	agents (self, MS)	agents (self, MS)
variables	✅	✅
can deploy to external services (AWS, GCP)	✅	✅

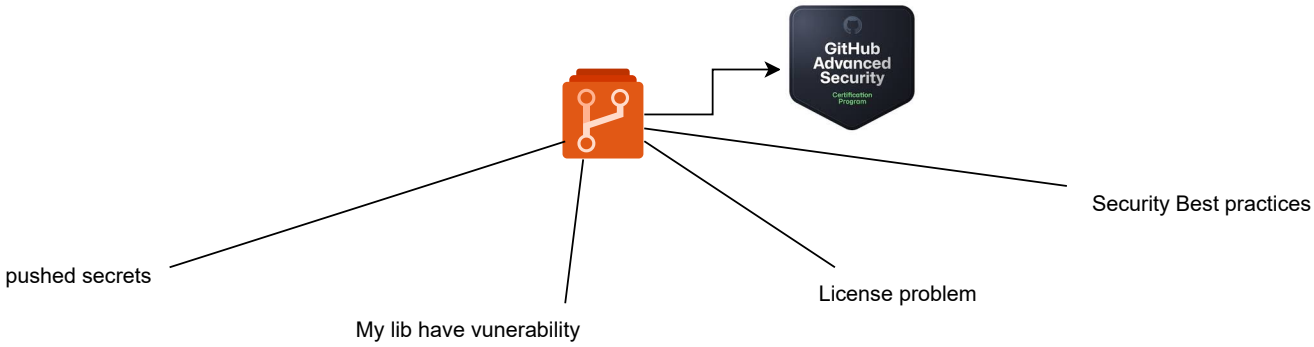
.github/workflows/your-yaml-actions-file

```yaml
trigger:
- conflict

pool:
  vmImage: ubuntu-latest

steps:
- script: echo Hello, world!
  displayName: 'Run a one-line script'

- script: |
    echo Add other tasks to build, test, and deploy your project.
    echo See https://aka.ms/yaml
  displayName: 'Run a multi-line script'
```


```yaml
# this is GIthub actions to help compare Azure DevOps YAML

name: Comparing with Azure Pipeline

#  trigger:
#   - conflict
on:
# ALLOW TO RUN MANUALLY .... ELSE not possible
  workflow_dispatch: {}

  # RUN WHEN CHNAGE HAPPENS IN CONFLICT BRANCH
  push:
    branches:
      - conflict


# # pool:
#    vmImage: ubuntu-latest

jobs:
   first:
     runs-on: ubuntu-latest

     steps:
     - name: One-line
       run: echo Hello, World !!!

     - name: Multi-Line
       run: |
         echo Add other tasks to build, test, and deploy your project.
         echo This is AJ !!!
```

Allow other identities to impersonate this application by establishing a trust with an external OpenID Connect (OIDC) identity provider. This federation allows you to get tokens to access Microsoft Entra ID protected resources that this application has access to like Azure and Microsoft graph

DevSecOps, which stands for development, security, and operations, is a framework that integrates security into all phases of the software development lifecycle. Organizations adopt this approach to reduce the risk of releasing code with security vulnerabilities.

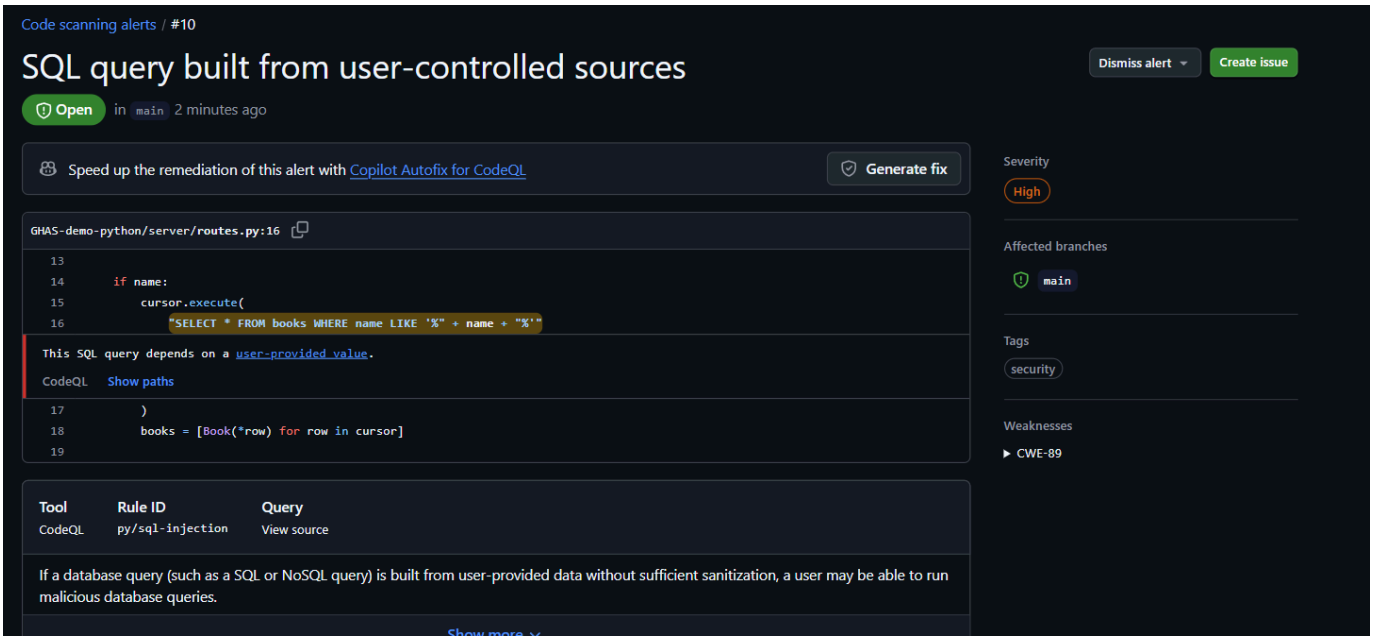


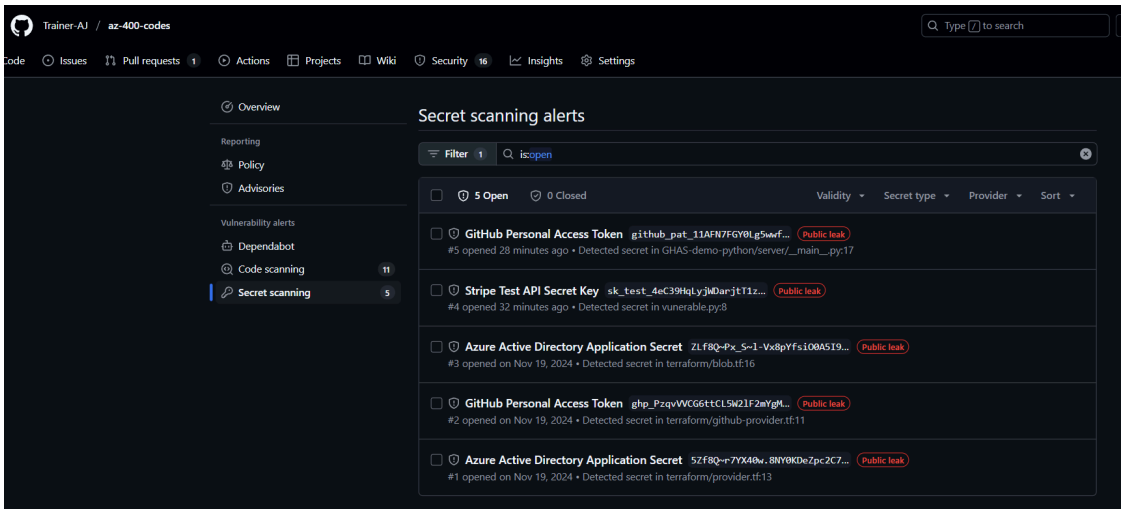
Secret Protection
Receive alerts on GitHub for detected secrets, keys, or other tokens. Disable

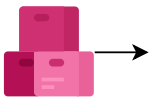
GitHub will always send alerts to partners for detected secrets in public repositories. Learn more about partner patterns.

Push protection
Block commits that contain supported secrets.

Github advance security => scan my code ==> highlighted problem ==> auto fix available



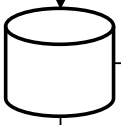




Internal Lib → approved and trusted lib with

upstream source

Source Code Repository



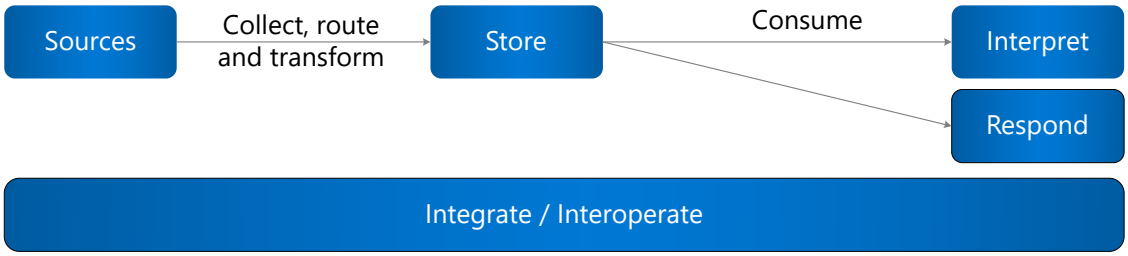
Feed

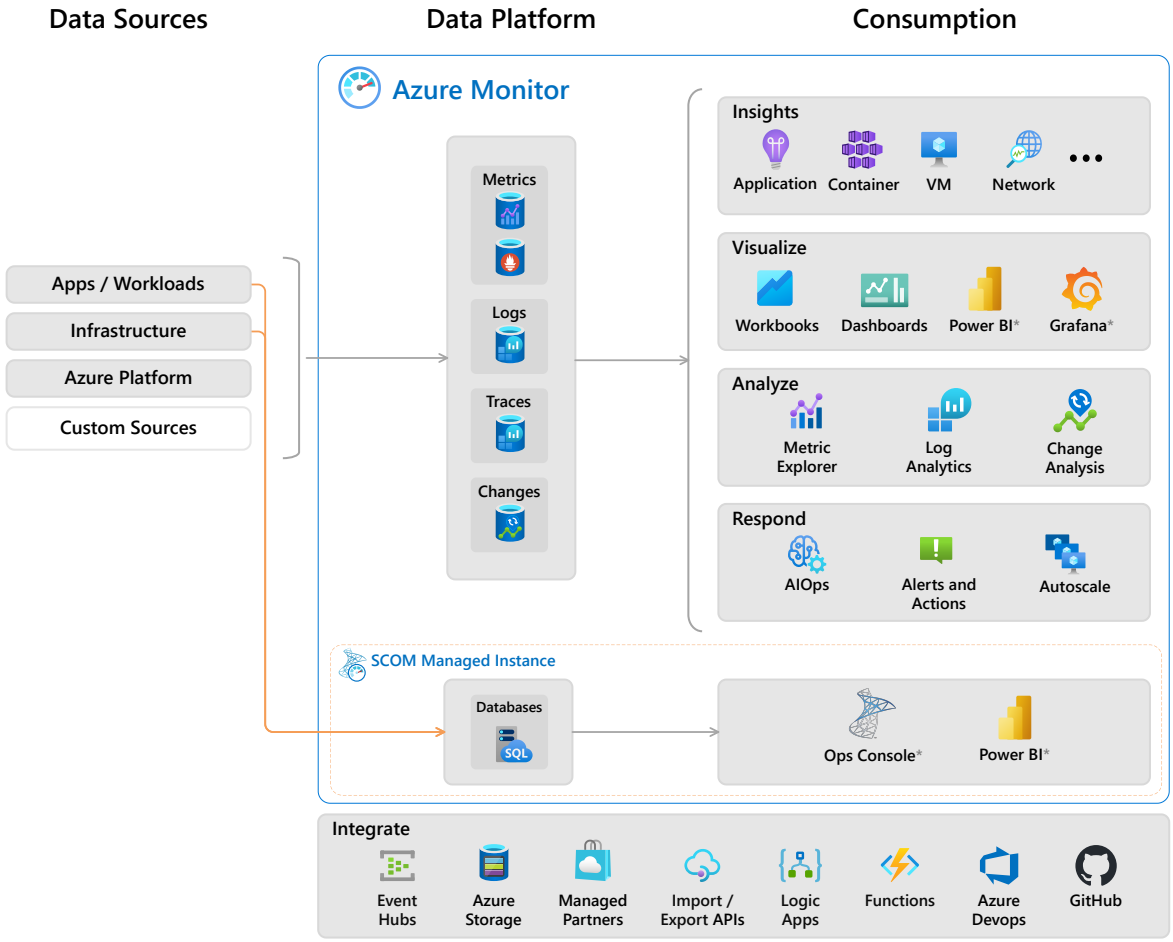
Azure Artifacts provides developers with a streamlined way to manage all their dependencies from a single feed. These feeds serve as repositories for storing, managing, and sharing packages, whether within your team, across organizations, or publicly online.

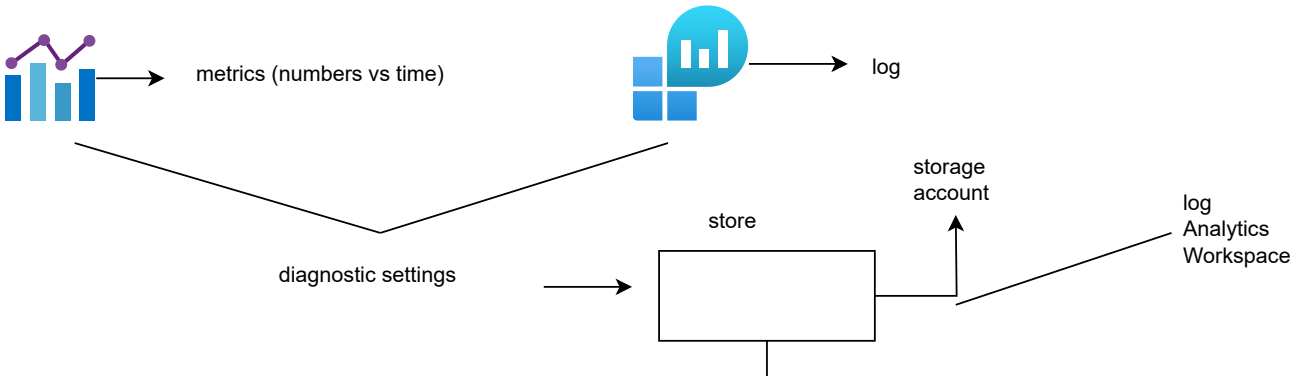
Azure Artifacts supports multiple package types, including NuGet, npm, Python, Maven, Cargo, and Universal Packages.

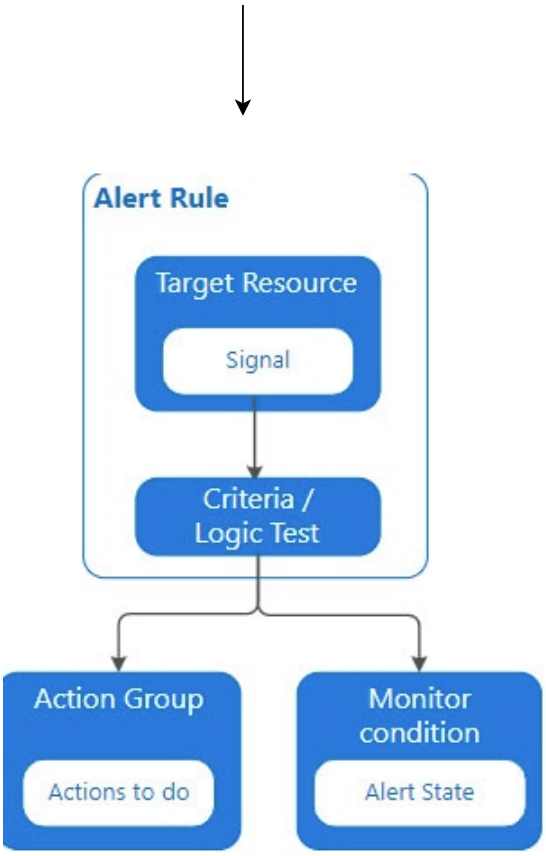
Azure Artifacts feeds are organizational constructs that enable you to store, manage, and share your packages while maintaining access control.

```
trigger:
  - main


pool:
  # Additional hosted image options are available: https://learn.microsoft.com/en-
us/azure/devops/pipelines/agents/hosted#software
  vmImage: ubuntu-latest


steps:


  - task: AdvancedSecurity-Codeql-Init@1
    inputs:
      languages: "java"
      # Supported languages: csharp, cpp, go, java, javascript, python, ruby, swift
      # You can customize the initialize task: https://learn.microsoft.com/en-
us/azure/devops/pipelines/tasks/reference/advanced-security-codeql-init-v1?view=azure-pipelines
      # If you're using a self-hosted agent to run CodeQL, use `enableAutomaticCodeQLInstall` to
automatically use the latest CodeQL bits on your agent:
      enableAutomaticCodeQLInstall: true


#   Add your custom build steps here
# - Ensure that all code to be scanned is compiled (often using a `clean` command to ensure you're
building from a clean state).
# - Disable the use of any build caching mechanisms as this can interfere with CodeQL's ability to
capture all the necessary data during the build.
# - Disable the use of any distributed/multithreaded/incremental builds as CodeQL needs to monitor
executions of the compiler to construct an accurate representation of the application.
# - For dependency scanning, ensure you have a package restore step for more accurate results.


# If you had a Maven app:
#   - task: Maven@4
#     inputs:
#       mavenPomFile: 'pom.xml'
#       goals: 'clean package'
#       publishJUnitResults: true
#       testResultsFiles: '**/TEST-*.xml'
#       javaHomeOption: 'JDKVersion'
#       jdkVersionOption: '1.17'
#       mavenVersionOption: 'Default'


# Or a general script:
#   - script: |
#       echo "Run, Build Application using script"
#       ./location_of_script_within_repo/buildscript.sh


  - task: AdvancedSecurity-Dependency-Scanning@1 # More details on this task:
https://learn.microsoft.com/en-us/azure/devops/pipelines/tasks/reference/advanced-security-
dependency-scanning-v1?view=azure-pipelines


  - task: AdvancedSecurity-Codeql-Analyze@1 # More details on this task: https://learn.microsoft.com/en-
us/azure/devops/pipelines/tasks/reference/advanced-security-codeql-analyze-v1?view=azure-pipelines
```


Sources
Collect, route and transform
Store
Consume
Interpret
Respond
Integrate / Interoperate


Data Sources
Data Platform
Consumption
Azure Monitor
Apps / Workloads
Infrastructure
Azure Platform
Custom Sources
Metrics
Logs
Traces
Changes
Insights
Application
Container
VM
Network
Visualize
Workbooks
Dashboards
Power BI*
Grafana*
Analyze
Metric Explorer
Log Analytics
Change Analysis
Respond
AIOps
Alerts and Actions
Autoscale
SCOM Managed Instance
Databases
SQL
Ops Console*
Power BI*
Integrate
Event Hubs
Azure Storage
Managed Partners
Import / Export APIs
Logic Apps
Functions
Azure Devops
GitHub


metrics (numbers vs time)
log
diagnostic settings
store
storage account
log Analytics Workspace



Service Hooks
Integrate with your favorite services by notifying them when events happen in your project.