PlantUML example:

Azure deployment:

@startuml

footer Kubernetes Plant-UML

scale max 1024 width

skinparam linetype polyline

skinparam nodesep 10

skinparam ranksep 10

' Azure

!define AzurePuml https://raw.githubusercontent.com/RicardoNiepel/Azure-PlantUML/release/2-1/dist

!includeurl AzurePuml/AzureCommon.puml

!includeurl AzurePuml/AzureSimplified.puml

!includeurl AzurePuml/Compute/AzureAppService.puml

!includeurl AzurePuml/Compute/AzureBatch.puml

!includeurl AzurePuml/Containers/AzureContainerRegistry.puml

!includeurl AzurePuml/Containers/AzureKubernetesService.puml

!includeurl AzurePuml/Databases/AzureDatabaseForPostgreSQL.puml

!includeurl AzurePuml/Databases/AzureCosmosDb.puml

!includeurl AzurePuml/Databases/AzureSqlDatabase.puml

!includeurl AzurePuml/DevOps/AzurePipelines.puml

!includeurl AzurePuml/Identity/AzureActiveDirectory.puml

!includeurl AzurePuml/Networking/AzureLoadBalancer.puml

!includeurl AzurePuml/Security/AzureKeyVault.puml

!includeurl AzurePuml/Storage/AzureBlobStorage.puml

!includeurl AzurePuml/Storage/AzureStorage.puml

' Kubernetes

!define KubernetesPuml https://raw.githubusercontent.com/dcasati/kubernetes-PlantUML/master/dist

!includeurl KubernetesPuml/kubernetes\_Context.puml

!includeurl KubernetesPuml/kubernetes\_Simplified.puml

!includeurl KubernetesPuml/OSS/KubernetesApi.puml

!includeurl KubernetesPuml/OSS/KubernetesIng.puml

!includeurl KubernetesPuml/OSS/KubernetesPod.puml

actor "DevOps" as devopsAlias

collections "Client Apps" as clientalias

collections "Helm Charts" as helmalias

left to right direction

' Azure Components

AzureActiveDirectory(aad, "\nAzure\nActive Directory", "Global")

AzureContainerRegistry(acr, "ACR", "Canada Central")

AzureCosmosDb(cosmos, "\nCosmos DB", "Global")

AzureKeyVault(keyvault, "\nAzure\nKey Vault", "Global")

AzureLoadBalancer(alb, "\nLoad\nBalancer", "Canada Central")

AzureSqlDatabase(sql, "\nExternal\ndata stores", "Canada Central")

AzurePipelines(ado, "CI/CD\nAzure Pipelines", "Global")

' Kubernetes Components

Cluster\_Boundary(cluster, "Kubernetes Cluster") {

KubernetesApi(KubernetesApi, "Kubernetes API", "")

Namespace\_Boundary(nsFrontEnd, "Front End") {

KubernetesIng(ingress, "API Gateway", "")

}

Namespace\_Boundary(nsBackEnd, "Back End") {

KubernetesPod(KubernetesBE1, "", "")

KubernetesPod(KubernetesBE2, "", "")

KubernetesPod(KubernetesBE3, "", "")

}

Namespace\_Boundary(nsUtil, "Utiliy Services") {

KubernetesPod(KubernetesUtil1, "", "")

KubernetesPod(KubernetesUtil2, "","")

}

}

Rel(devopsAlias, aad, "AUTH")

Rel(helmalias, KubernetesApi, "helm upgrade")

Rel(aad, keyvault, " ")

Rel(KubernetesApi, aad, "RBAC", "ASYNC")

Rel(clientalias, alb, "HTTP", "ASYNC")

Rel(alb, ingress, "HTTP", "ASYNC")

Rel(ingress, KubernetesBE1, " ")

Rel(KubernetesBE1, KubernetesBE2, " ")

Rel(KubernetesBE1, KubernetesBE3, " ")

Rel(KubernetesBE2, sql, " ")

Rel(KubernetesBE3, keyvault, "Pod Identity")

Rel(KubernetesBE3, cosmos, " ")

Rel(ado, acr, "docker push")

Rel\_U(KubernetesApi, acr, "docker pull")

@enduml

GCP example:

@startuml

!define GCPPuml https://raw.githubusercontent.com/Crashedmind/PlantUML-icons-GCP/master/dist

!include GCPPuml/GCPCommon.puml

!include GCPPuml/Compute/Cloud\_Functions.puml

!include GCPPuml/Networking/Cloud\_Firewall\_Rules.puml

!include GCPPuml/Compute/Compute\_Engine.puml

!include GCPPuml/Storage/Cloud\_Storage.puml

/'

The other icons will need to come from other stdlib libraries: backup, users, clients.

'/

!include <awslib/AWSCommon>

!include <awslib/AWSSimplified.puml>

!include <awslib/Compute/all.puml>

!include <awslib/mobile/all.puml>

!include <awslib/general/all.puml>

Users(Users, "Friends", " ")

Mobile(Mobile, "", " ")

Client(Client, "Kid / Owner", " ")

Client(ClientMinecraft, "", " ")

Cloud\_Functions(Cloud\_FunctionsStart, "Start Server", "Cloud Functions")

Cloud\_Functions(Cloud\_FunctionsStop, "Stop Server", "Cloud Functions")

Cloud\_Functions(Cloud\_FunctionAdd, "Add a Friend", "Cloud Functions")

Compute\_Engine(Compute\_Engine, "MineCraft Server", "Compute Engine")

Cloud\_Storage(Cloud\_Storage, "MineCraft Backups", "Cloud Storage")

Cloud\_Firewall\_Rules(Cloud\_Firewall\_Rules\_Starter,"Minecraft Backups", "Cloud Firewall Rules")

Cloud\_Firewall\_Rules(Cloud\_Firewall\_Rules\_Friend,"Minecraft Backups", "Cloud Firewall Rules")

@enduml

AWS example:

@startuml

!include <awslib/AWSCommon>

!include <awslib/AWSSimplified.puml>

!include <awslib/Compute/all.puml>

!include <awslib/mobile/all.puml>

!include <awslib/general/all.puml>

!include <awslib/GroupIcons/all.puml>

skinparam linetype polyline

' skinparam linetype ortho

package "AWS Cloud" {

EC2(Smadex, "Smadex Service", " ")

}

Users(Users, "Users", " ")

TraditionalServer(AdExchange, "Ad Exchange", " ")

Mobile(Mobile, "Publisher app or web", " ")

Users -down-> Mobile: 1. Visits

Mobile -right-> AdExchange: 2. Start auction

AdExchange -right-> Smadex: 3. Bid request / response

Smadex -left-> Mobile: 4. Show Ad

Users -right-> Smadex: 5. Impression / click / install / event {request id}

@enduml

C4 example:

@startuml

!include <C4/C4\_Context.puml>

!include <office/Users/user.puml>

LAYOUT\_WITH\_LEGEND()

title Top level Context diagram for ACME Global Widget Production

Person(AcmeProdOwner, Acme Production Owner , "<$user> \n Responsible for Production of widgets in ACME Production Site" )

Person(3rdProdOwner, 3rdParty Production Owner , "<$user> \n Responsible for Production of widgets in 3rdParty Production Site" )

System\_Boundary(Remote, "Remote") {

System(ProdSystemHost, "Production Host", "Drives widget Production\n Analyses widget Production data to create reports")

System(Analytics, "Analytics", "Provides data analysis and a dashboard view data - including relevant Production data")

System(Monitoring, "Monitoring", "Monitors Production Host\nUses AWS Services")

System\_Ext(SupplyChain, "SupplyChain", "Provides parts to make the widgets")

System\_Ext(InventoryTracking, "InventoryTracking", "Customer Reference")

}

Enterprise\_Boundary(AcmeProd, "ACME Production Site") {

System(AcmeWorkStation1, "WorkStation1", " Production setup for a group of widgets")

System(AcmeWorkStation2, "WorkStation2", " Production setup for a group of widgets")

System(AcmeWorkStationN, "WorkStationN", " Production setup for a group of widgets")

}

Enterprise\_Boundary(3rdProd, "3rdParty Production Site") {

System(3rdWorkStation1, "WorkStation1", " Production setup for a group of widgets")

System(3rdWorkStation2, "WorkStation2", " Production setup for a group of widgets")

System(3rdWorkStationN, "WorkStationN", " Production setup for a group of widgets")

}

Rel\_U(AcmeWorkStation1, ProdSystemHost, "Sends Production report for widget")

Rel\_U(AcmeWorkStation2, ProdSystemHost, "Sends Production report for widget")

Rel\_U(AcmeWorkStationN, ProdSystemHost, "Sends Production report for widget")

Rel\_U(3rdWorkStation1, ProdSystemHost, "Sends Production report for widget")

Rel\_U(3rdWorkStation2, ProdSystemHost, "Sends Production report for widget")

Rel\_U(3rdWorkStationN, ProdSystemHost, "Sends Production report for widget")

Rel\_D(AcmeProdOwner, ProdSystemHost, "Reviews Acme and 3rdParty site Production reports")

Rel\_D(3rdProdOwner, ProdSystemHost, "Reviews 3rdParty site Production reports")

Rel\_D(Analytics, ProdSystemHost, "Data Analysis")

Rel\_D(SupplyChain, ProdSystemHost, "Parts")

Rel\_D(InventoryTracking, ProdSystemHost, "Tracking")

Rel\_D(Monitoring, ProdSystemHost, "Monitoring")

footer %filename() rendered with PlantUML version %version()\nThe Hitchhiker’s Guide to PlantUML

@enduml

WBS diagram example:

@startwbs

\* Business Process Modelling WBS

\*\* Launch the project

\*\*\* Complete Stakeholder Research

\*\*\* Initial Implementation Plan

\*\* Design phase

\*\*\* Model of AsIs Processes Completed

\*\*\*\* Model of AsIs Processes Completed1

\*\*\*\* Model of AsIs Processes Completed2

\*\*\* Measure AsIs performance metrics

\*\*\* Identify Quick Wins

\*\* Complete innovate phase

@endwbs

MindMap example

@startmindmap

\* Debian

\*\* Ubuntu

\*\*\* Linux Mint

\*\*\* Kubuntu

\*\*\* Lubuntu

\*\*\* KDE Neon

\*\* LMDE

\*\* SolydXK

\*\* SteamOS

\*\* Raspbian with a very long name

\*\*\* <s>Raspmbc</s> => OSMC

\*\*\* <s>Raspyfi</s> => Volumio

@endmindmap

Gantt chart example

@startgantt

[Prototype design] requires 15 days

[Test prototype] requires 10 days

-- All example --

[Task 1 (1 day)] requires 1 day

[T2 (5 days)] requires 5 days

[T3 (1 week)] requires 1 week

[T4 (1 week and 4 days)] requires 1 week and 4 days

[T5 (2 weeks)] requires 2 weeks

@endgantt

Deployment diagram example:

@startuml

node node1

node node2

node node3

node node4

node node5

node1 -- node2 : label1

node1 .. node3 : label2

node1 ~~ node4 : label3

node1 == node5

@enduml