

THE DOWNLOAD

TECH TALKS BY HPCC SYSTEMS

Episode
35



Brought to you by the HPCC Systems Developer Community

August 12, 2020

Welcome!

- Please share: Let others know you are here with [#HPCCTechTalks](#)
- Ask questions! We will answer as many questions as we can following each speaker.
- Look for polls at the bottom of your screen. Exit full-screen mode or refresh your screen if you don't see them.
- We welcome your feedback - please rate us before you leave today.
- Visit our [website](#) for information after the event.
- Want to be one of our featured speakers? Let us know! techtalks@hpccsystems.com



Jessica Lorti

Director, Marketing

LexisNexis Risk Solutions

Jessica.Lorti@lexisnexisrisk.com



HPCC SYSTEMS®

Community announcements

Platform updates:

- 7.10.8 Gold – [Download Now](#) / Get the [Release Notes](#)
- Test our new Cloud native version. Find out everything you need to know here.
- [HPCC Systems Roadmap](#)

Read the [latest blogs](#) on the community portal:

- [HPCC Systems COVID-19 Tracker: Is there a better way to tell the story?](#)
- [Securing your HPCC Systems environment and protecting your data](#)
- Intern project updates from [Matthias Murray](#) and [Nathan Halliday](#)

Catch up on our Summer Tech Talk Interview Series

- [Richard Chapman](#) **Coming Soon...**
- [Jake Smith](#)
- [Gavin Halliday](#)
- [Shamser Ahmed](#)
- [Ming Wang](#)
- Godson Fortil
- Anthony Fishbeck
- Lorraine Chapman

Recent News and Events

- Call for Presentations and Poster Abstracts – [Deadline this Friday August 14](#)
- Virtual poster presentations competition for students working on HPCC Systems related projects. [See the rules](#) / [View past poster contests](#)



Lorraine Chapman

*Consulting Business Analyst &
Internship Program Manager*

LexisNexis® Risk Solutions

Lorraine.chapman@lexisnexisrisk.com

2020 HPCC Systems
Community Virtual Summit

Call for Presentations
Open through Aug 14

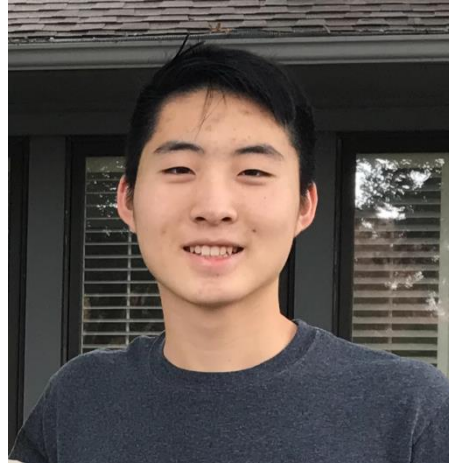


Today's speakers



Nathan Halliday

Nathan Halliday is a high school student who has just completed his A' Levels. He will start university later this year, having received an offer to study mathematics. He is interested in quantum computing, which may influence his future career path.



Jefferson "Jeff" Mao

High School Student
Lambert High School, Suwanee, GA
jeffs.mao@gmail.com

Jefferson is a 12th grade high school student who has his eyes set on studying business at the University of Pennsylvania in the future. He is already something of an entrepreneur having founded Philosophy Robotics LLC, a software company that produces software for resellers such as automated checkout services, reselling tools and web scraping applications. He heard about the HPCC Systems intern program when taking part in CodeDay Atlanta, where he was a Best in Show prize winner.



Jack Fields

High School Student
American Heritage School, Delray Beach, FL
jeffs.mao@gmail.com

Jack is a 12th grade high school student who has developed an impressive amount of experience in Java, Python and C++ from the robotics and computer science courses provided by his school and his involvement in the Stallion Robotics Team 5472. Jack is the Director of Programming for the team, currently working on an Autonomous Security Robot. Jack submitted a poster into our 2019 Technical Poster Contest showcasing the progress the team has on made on the robot.



THE DOWNLOAD

TECH TALKS BY HPCC SYSTEMS



HPCC SYSTEMS®

The Parallel Workflow Engine

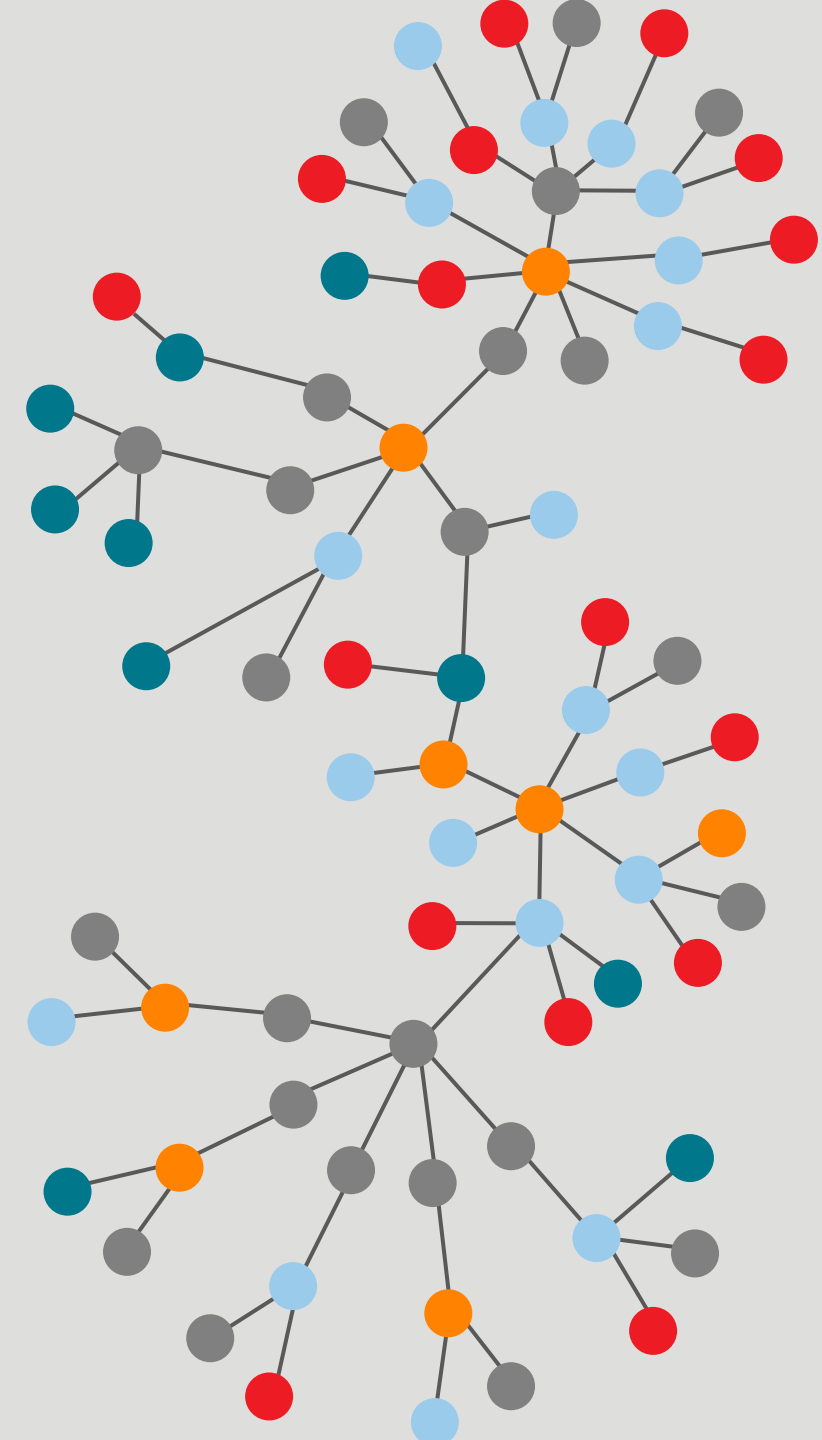
Nathan Halliday
High School Graduate



Quick poll:

Have you heard of the workflow, and its use when processing ECL queries?

See poll on bottom of presentation screen

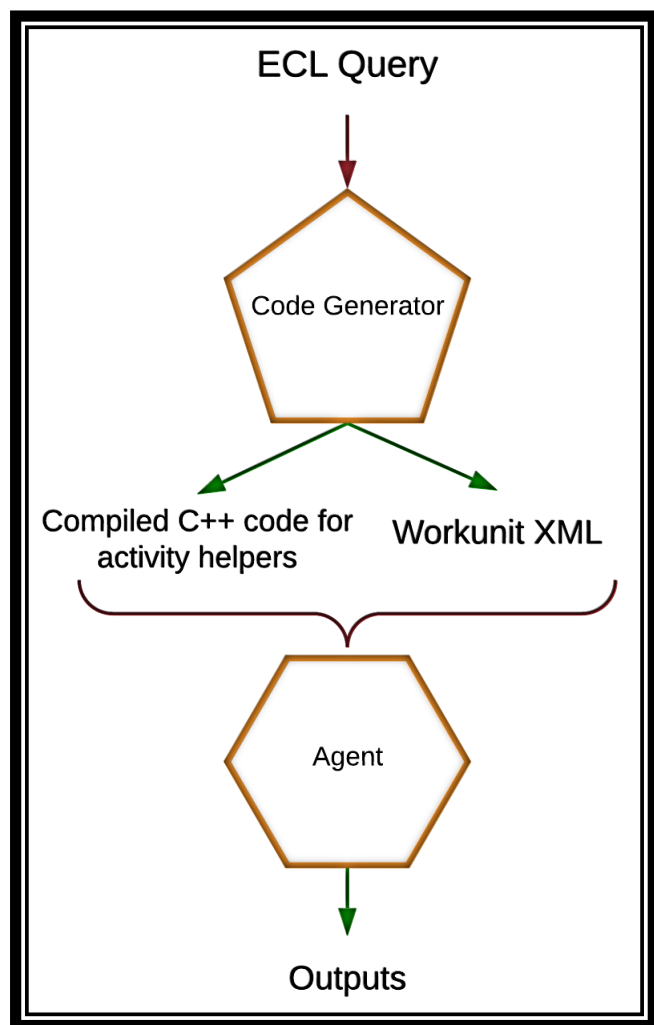


Presentation Overview

What is the
Workflow?

How does the
Parallel
Engine work?

General Flowchart for a Job



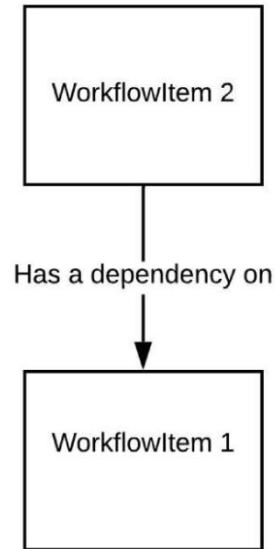
```
switch (wfid) {  
    case 2U: {  
        ctx->executeGraph("graph1",false,0,NULL);  
        ctx->setResultString(0,1U,5U,"Done!");  
    }  
    break;  
}
```

Helpers Workunit XML x

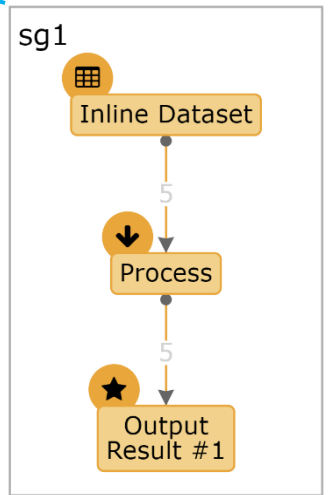
```
667 <Tracing>  
668 <EclAgentBuild>community_7  
669 </Tracing>  
670 <Workflow>  
671 <Item label="a"  
672     mode="normal"  
673     state="null"  
674     type="normal"  
675     wfid="1"/>  
676 <Item label="b"  
677     mode="normal"  
678     state="null"  
679     type="normal"  
680     wfid="2"/>
```


Workunit Item

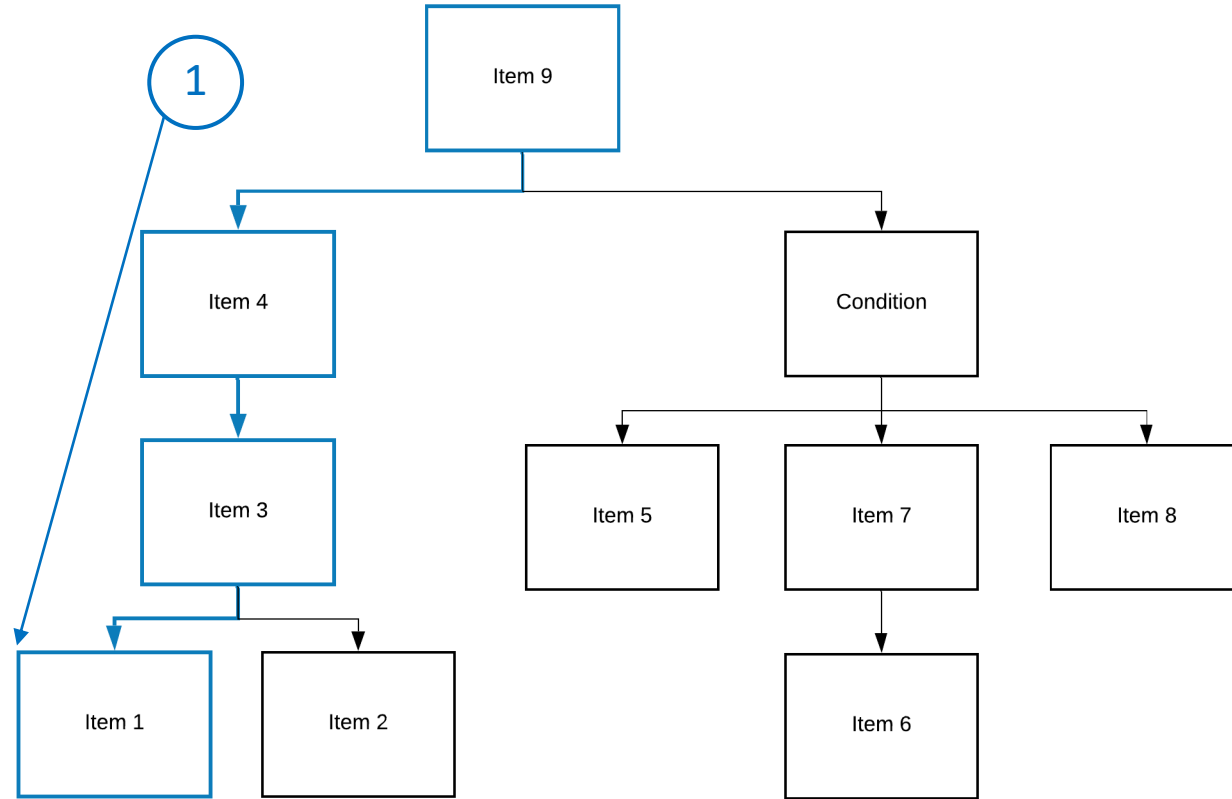
```
<Workflow>
  <Item .... wfid="1"/>
  <Item .... wfid="2">
    <Dependency wfid="1"/>
  </Item>
</Workflow>
```



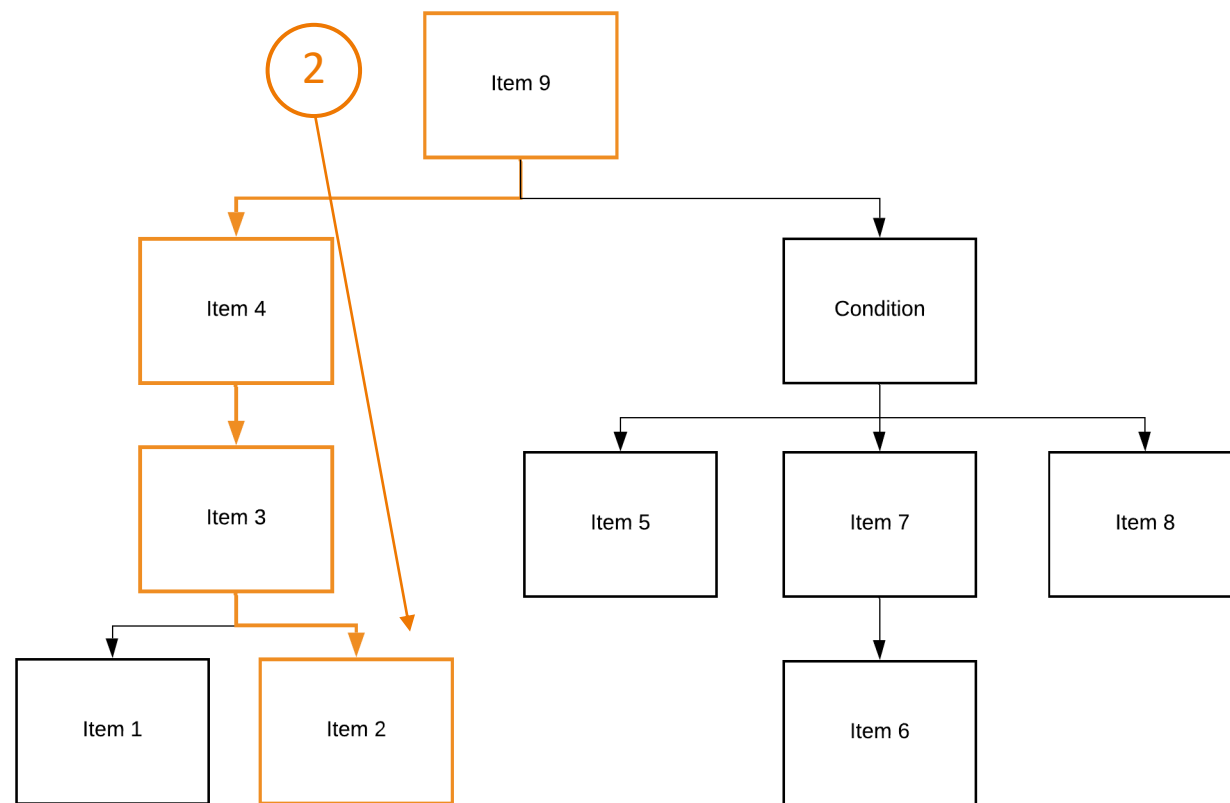
Item 1



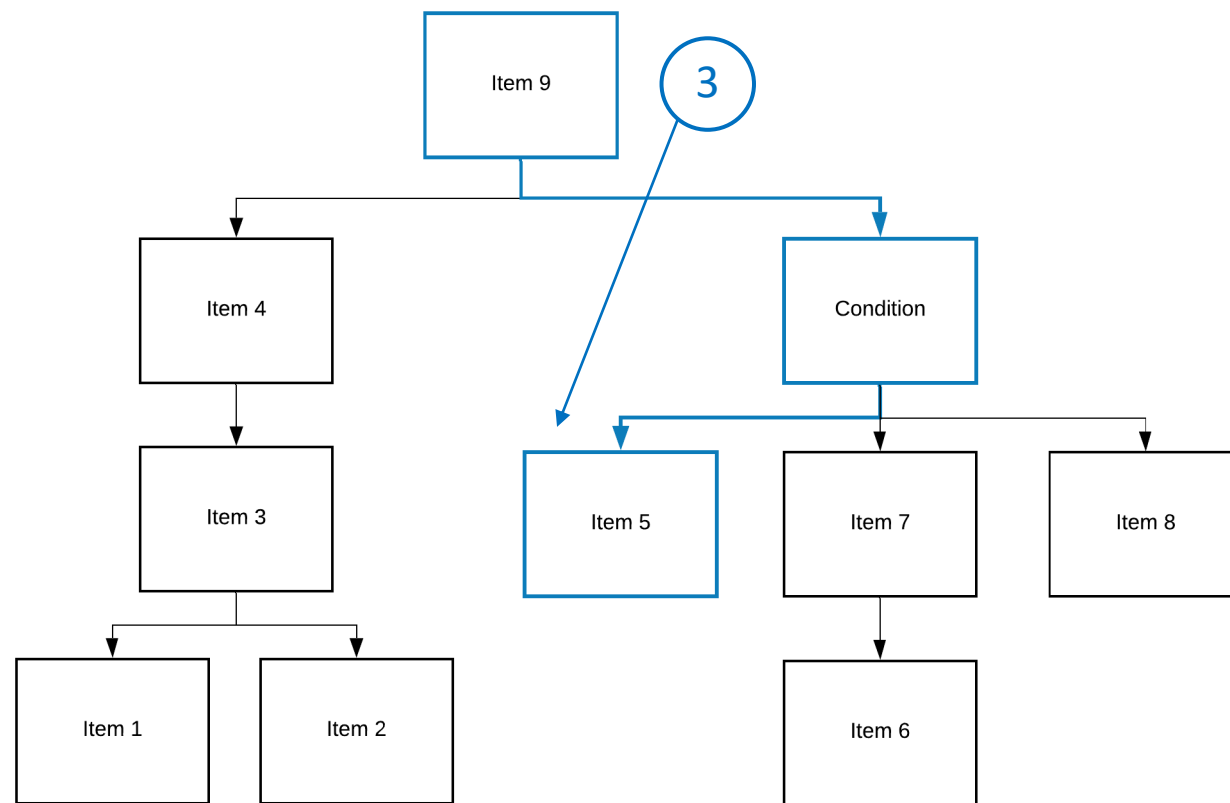
Sequential Workflow Engine



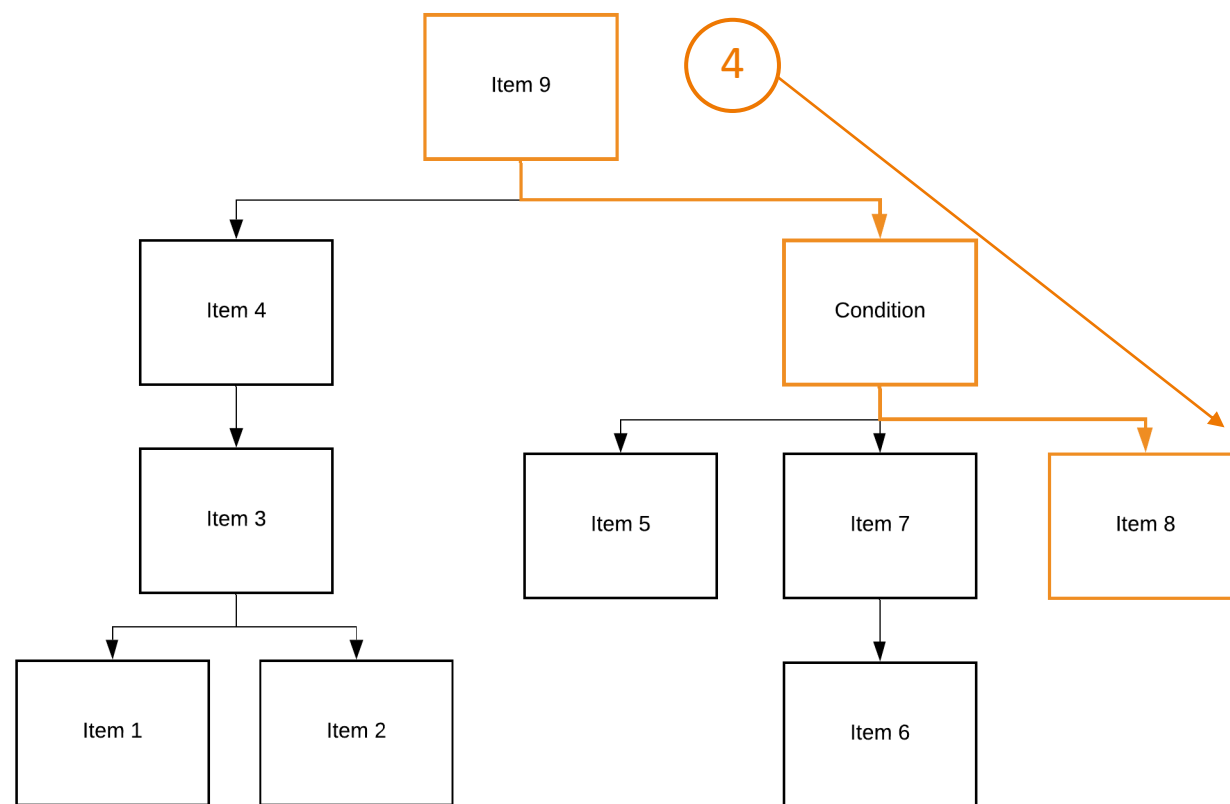
Sequential Workflow Engine



Sequential Workflow Engine

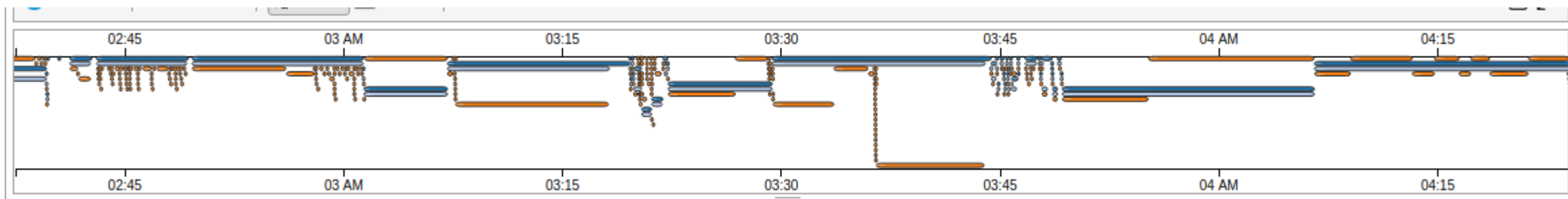


Sequential Workflow Engine

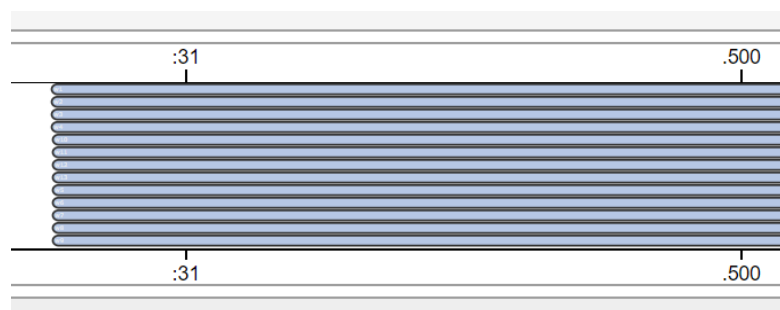


Workunit Gantt Chart

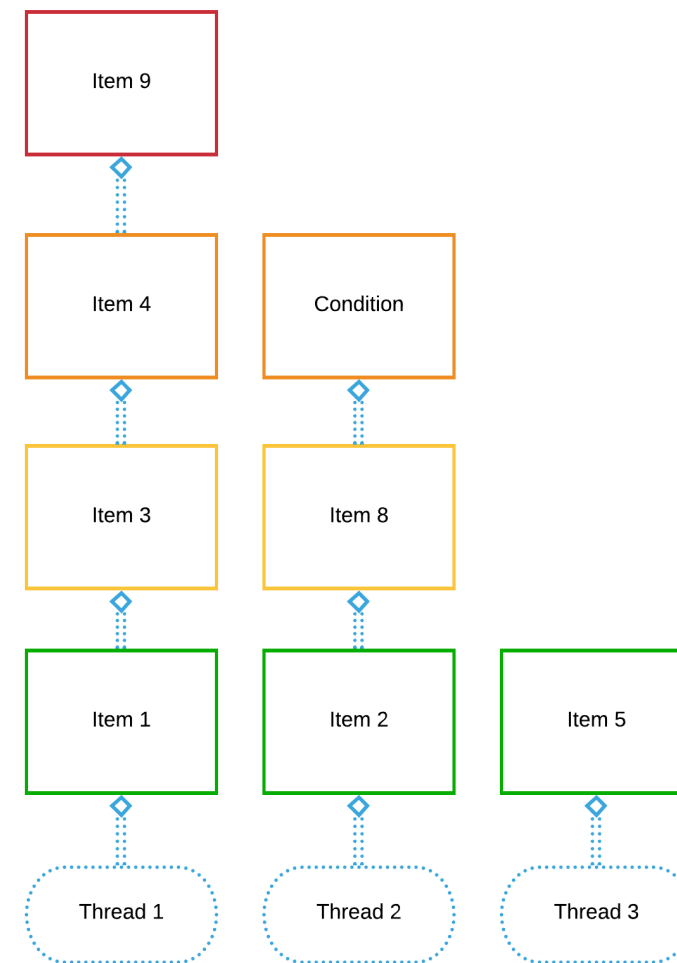
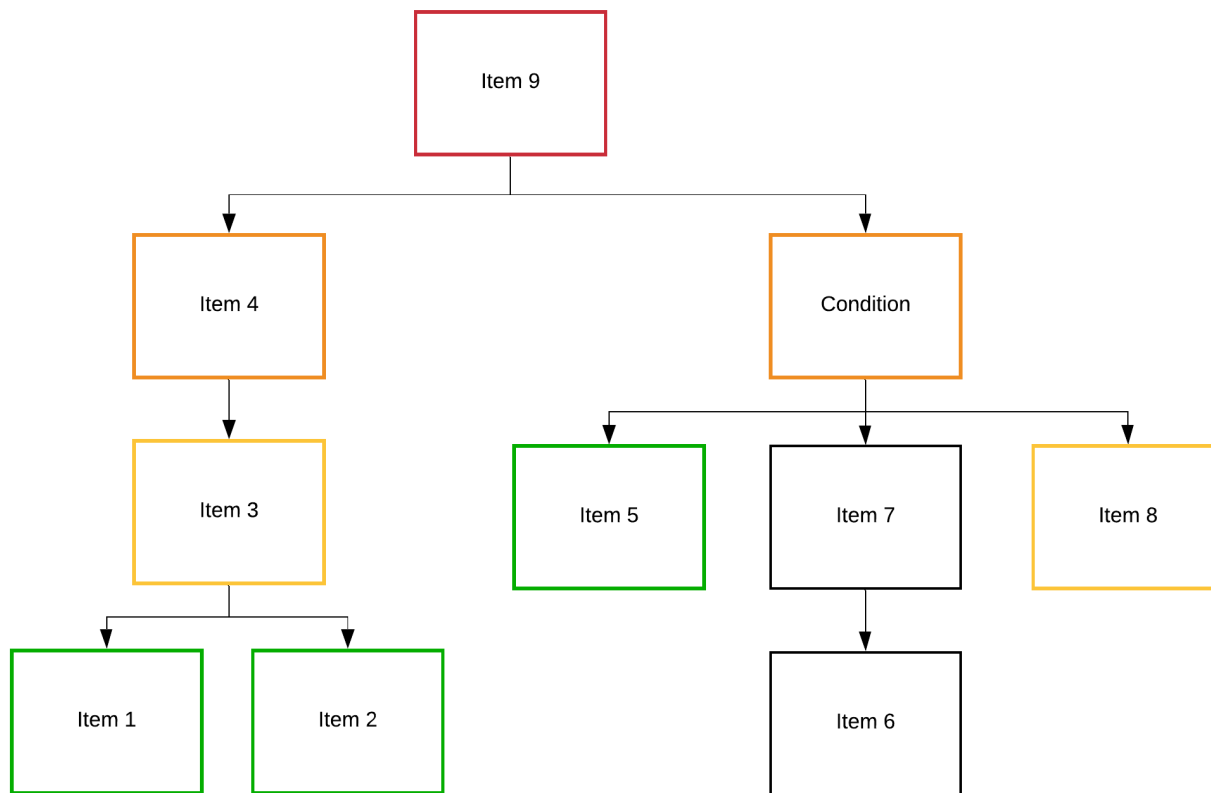
Sequential



Parallel

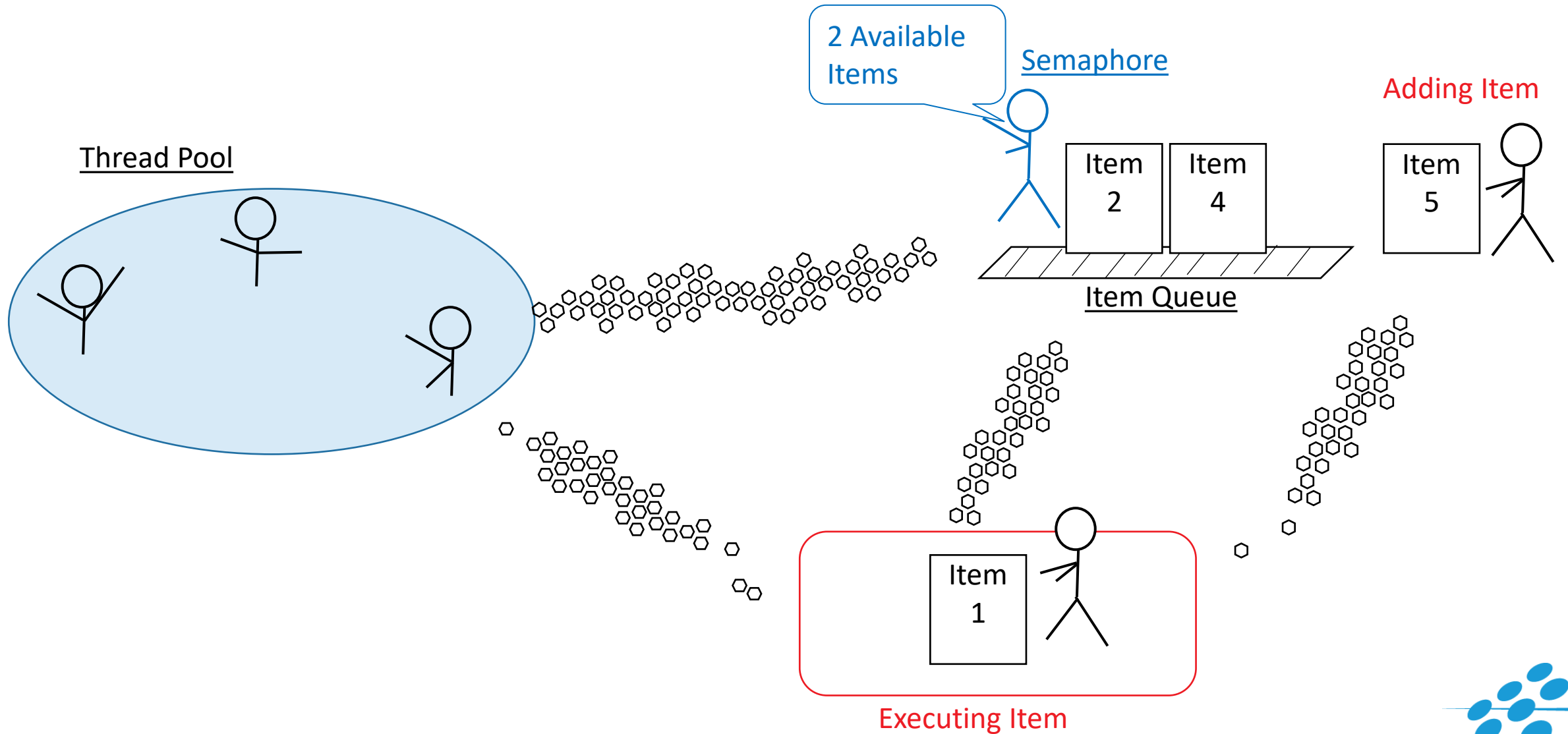


How could this be done in parallel?



Parallel Engine Implementation

The Parallel Engine



List of All Modes

```
EnumMapping wftypes[] =
{
    { WFTypeNormal, "normal" },
    { WFTypeSuccess, "success" },
    { WFTypeFailure, "failure" },
    { WFTypeRecovery, "recovery" },
    { WFTypeWait, "wait" },
    { WFTypeSize, NULL }
};

EnumMapping wfmodes[] =
{
    { WFModeNormal, "normal" },
    { WFModeCondition, "condition" },
    { WFModeSequential, "sequential" },
    { WFModeParallel, "parallel" },
    { WFModePersist, "persist" },
    { WFModeBeginWait, "bwait" },
    { WFModeWait, "wait" },
    { WFModeOnce, "once" },
    { WFModeCritical, "critical" },
    { WFModeSize, NULL }
};
```

High Priority	Lower Priority
Parallel	Recovery
Ordered	Critical, Once
Sequential	Wait
Success, Failure	
Condition	
Persist	

Development Timeline

- Research
 - Current workflow engine
 - Techniques for thread safe algorithms
 - Intel® Threading Building Blocks

- Design
- Implementation
- Testing



For each workflow mode

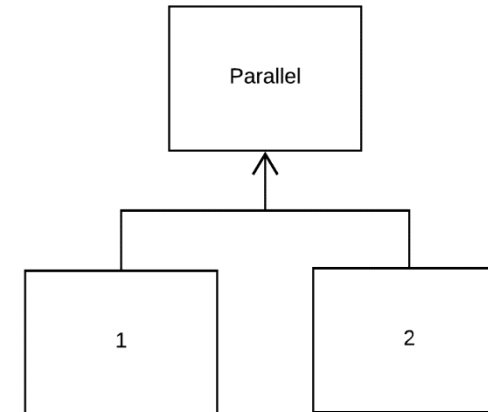
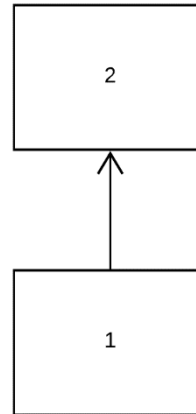
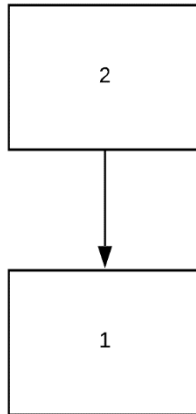
High Priority
Parallel
Ordered
Sequential
Success, Failure
Condition
Persist

Parallel

PARALLEL(action1, action2);

or

PARALLEL(action2, action1);



High Priority

Parallel

Ordered

Sequential

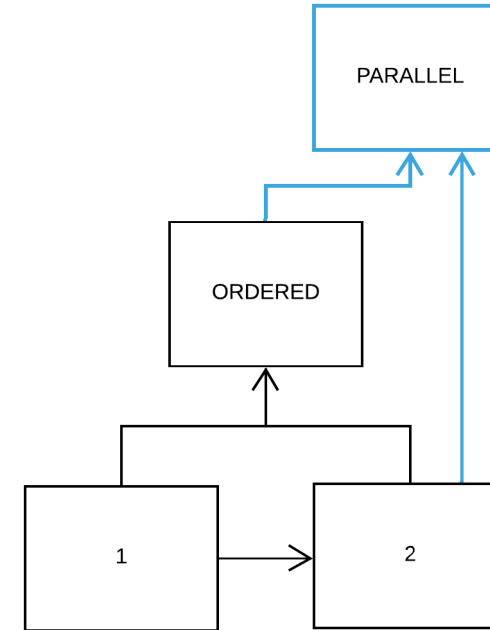
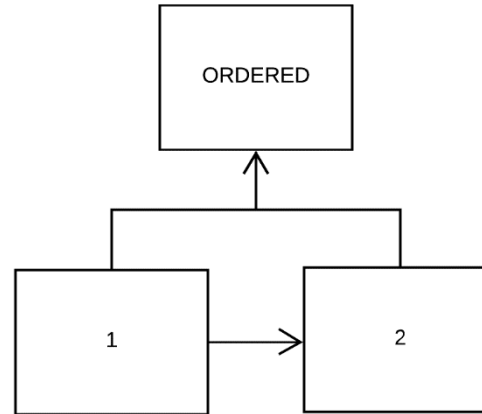
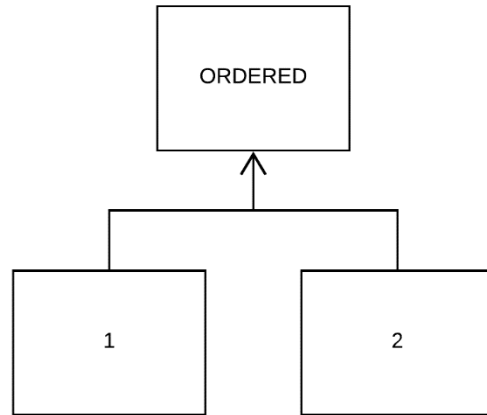
Success, Failure

Condition

Persist

Ordered

ORDERED(a1, a2);



PARALLEL(ORDERED(a1, a2), a2);

High Priority

Parallel

Ordered

Sequential

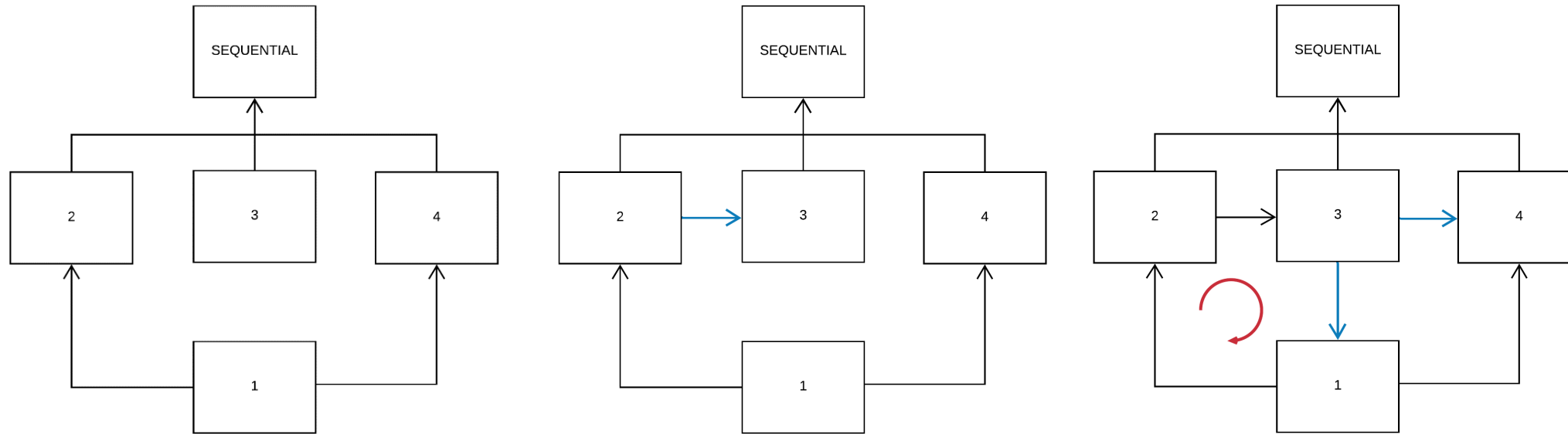
Success, Failure

Condition

Persist

Sequential

SEQUENTIAL(a2, a3, a4);



High Priority

Parallel

Ordered

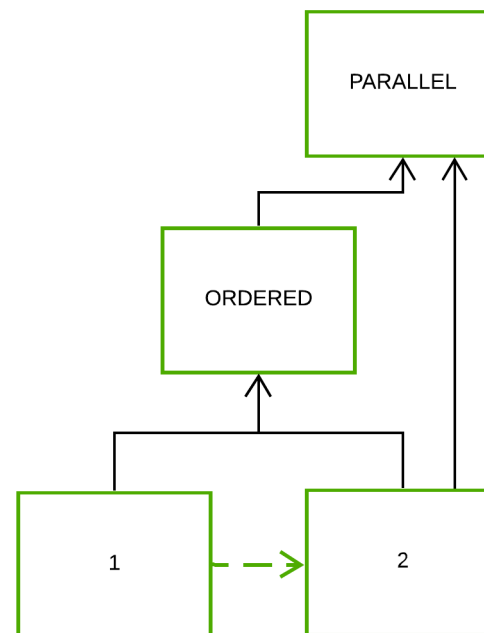
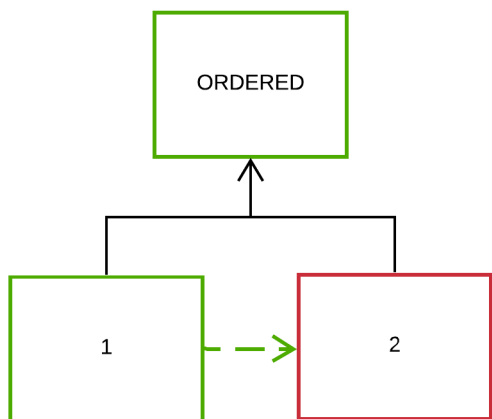
Sequential

Success, Failure

Condition

Persist

Ordered



High Priority

Parallel

Ordered

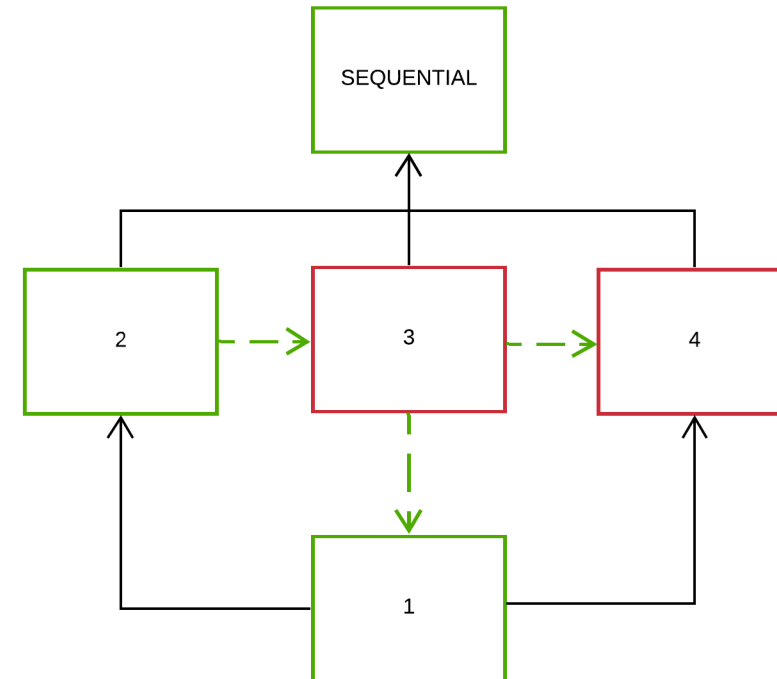
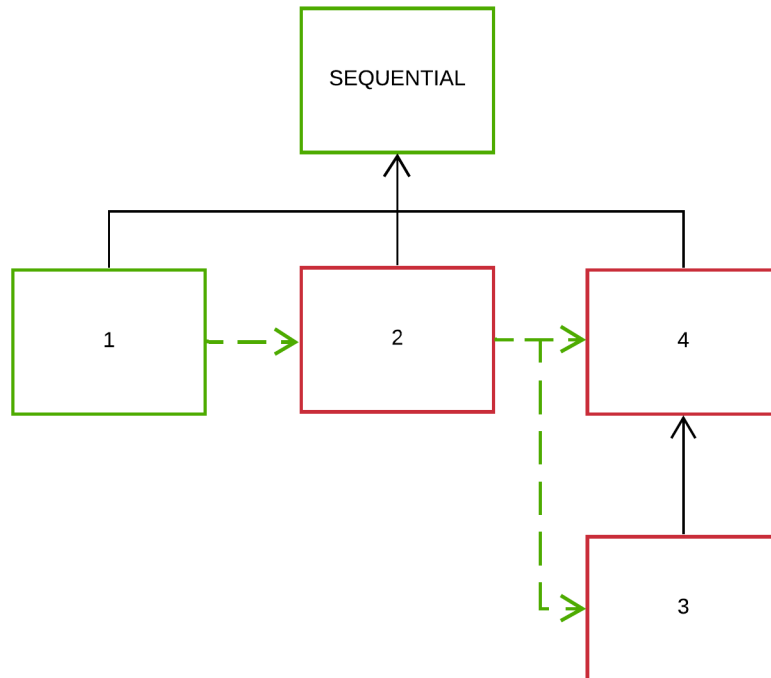
Sequential

Success, Failure

Condition

Persist

Sequential



High Priority

Parallel

Ordered

Sequential

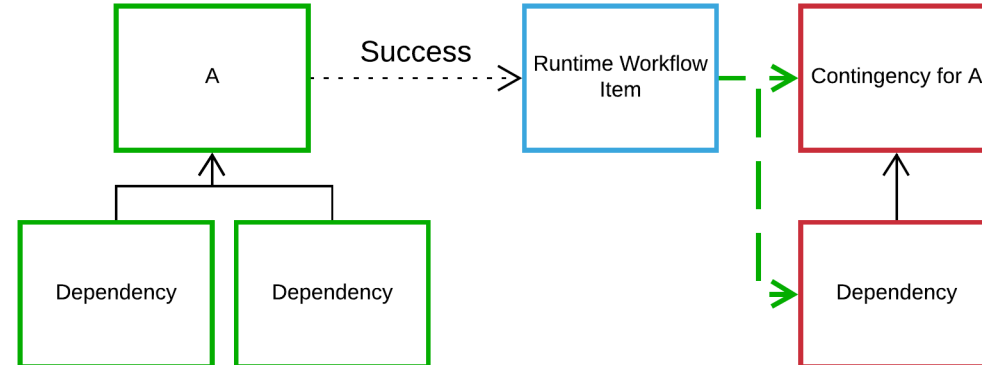
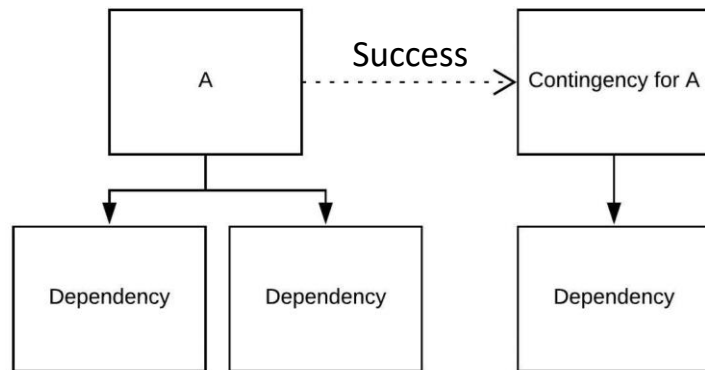
Success, Failure

Condition

Persist

Contingency Clauses

A : **SUCCESS**(A_Contingency);



High Priority

Parallel

Ordered

Sequential

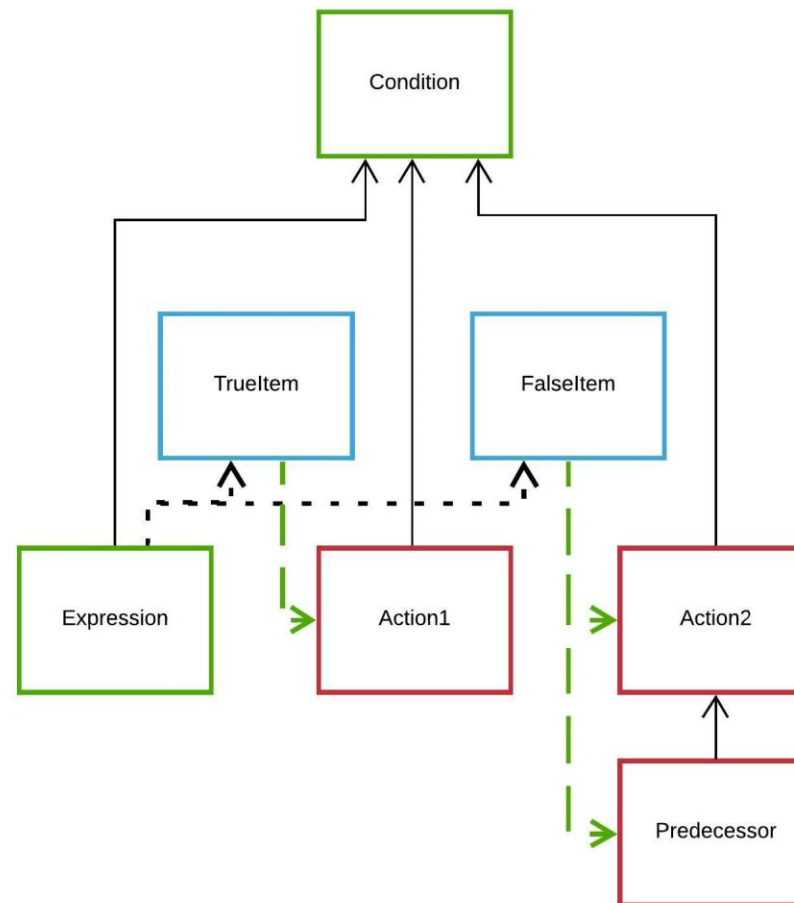
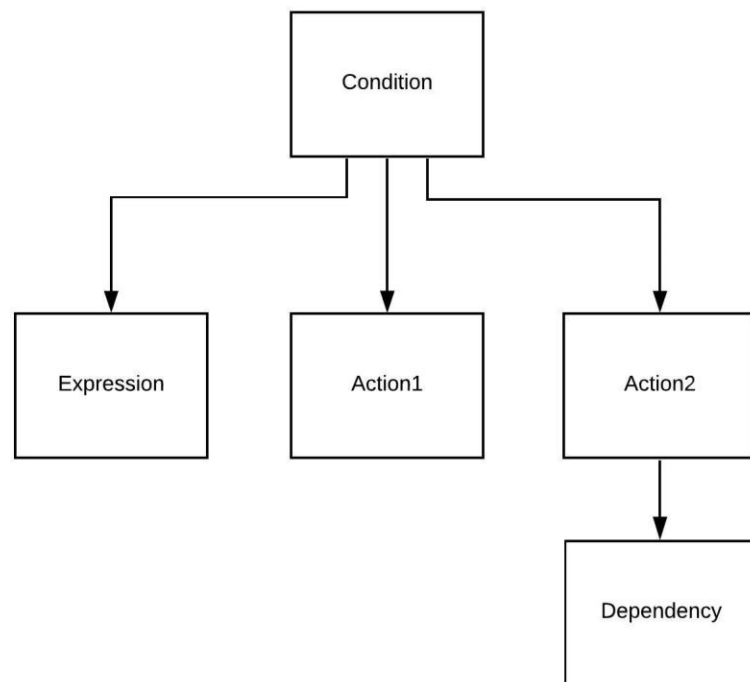
Success, Failure

Condition

Persist

IF Function

```
IF(expression, trueresult_a1, falseresult_a2);
```



High Priority

Parallel

Ordered

Sequential

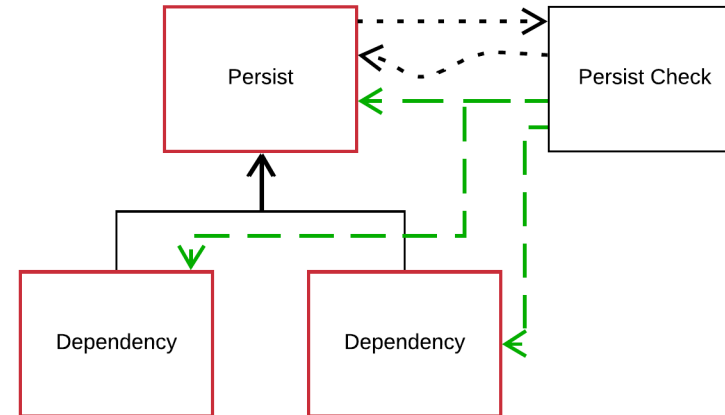
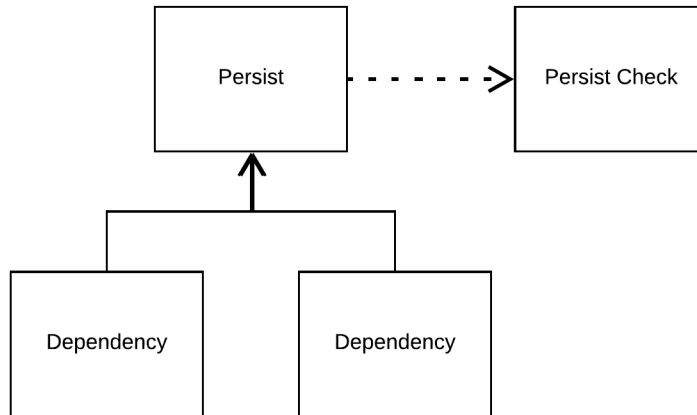
Success, Failure

Condition

Persist

Persist

```
attribute : PERSIST('thisPersist');
```



High Priority

Parallel

Ordered

Sequential

Success, Failure

Condition

Persist

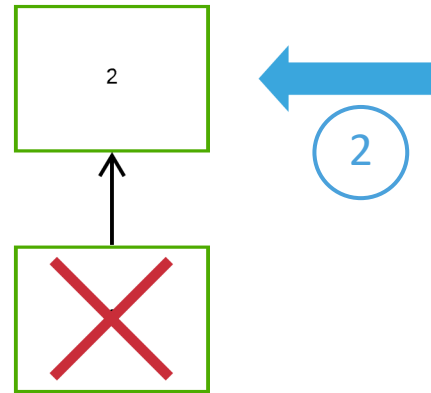
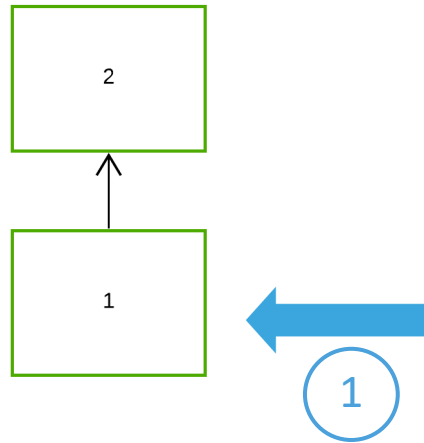
What happens when the workflow fails?



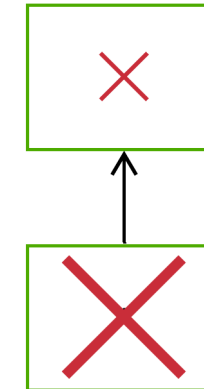
Workflow Failure

a1 := **FAIL**() : independent;

Starting State



Final State



High Priority

Parallel

Ordered

Sequential

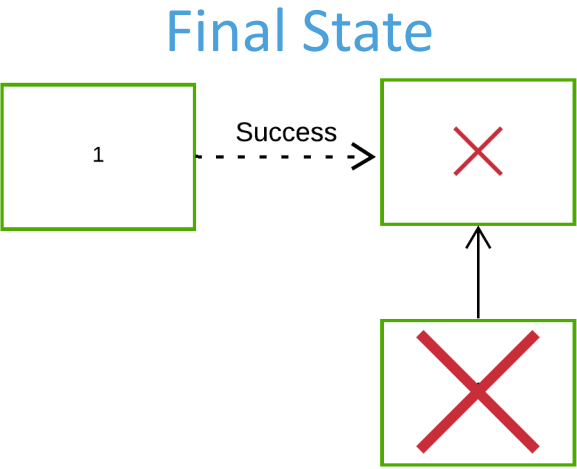
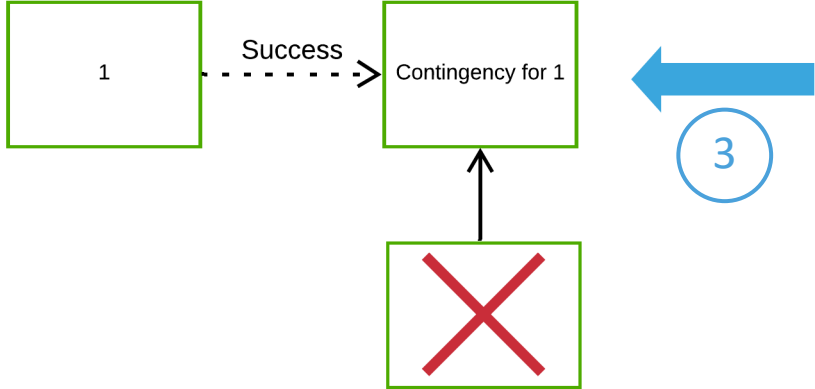
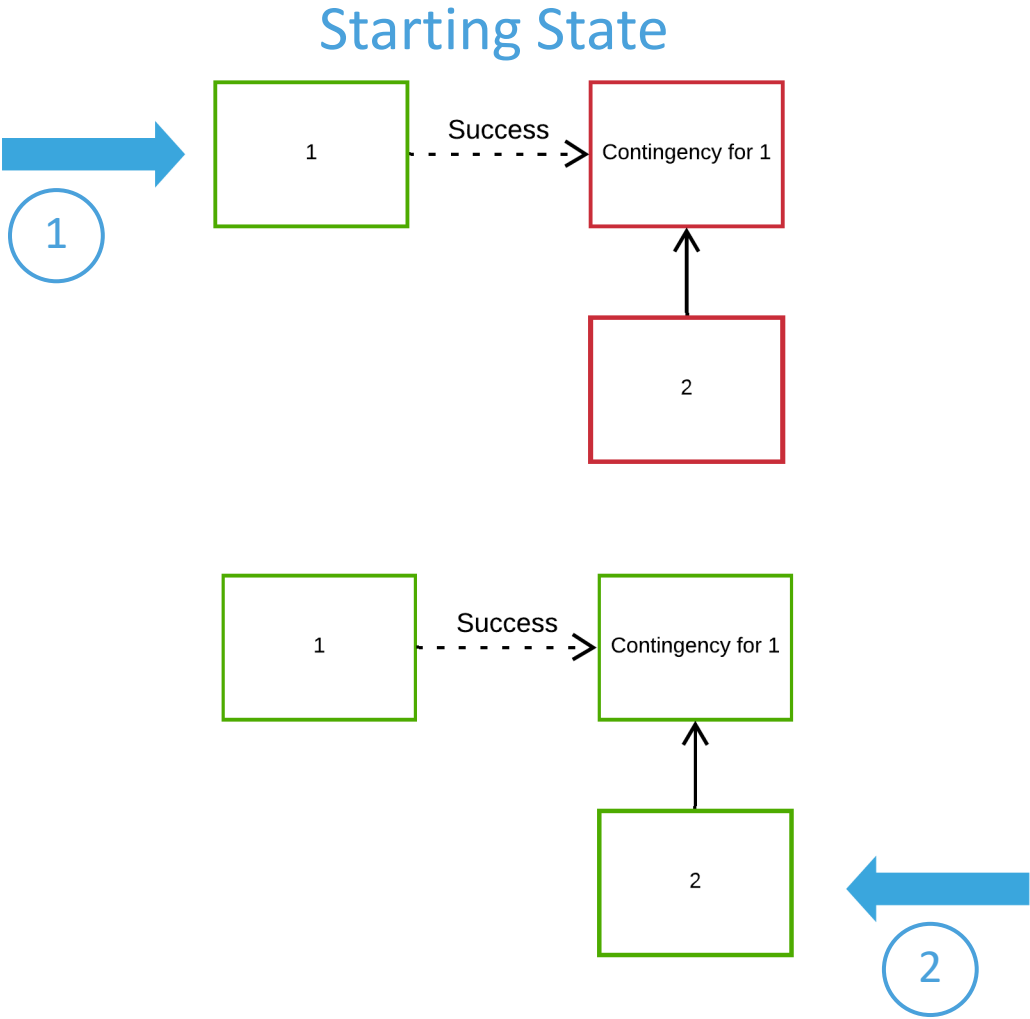
Success, Failure

Condition

Persist

Failed Items

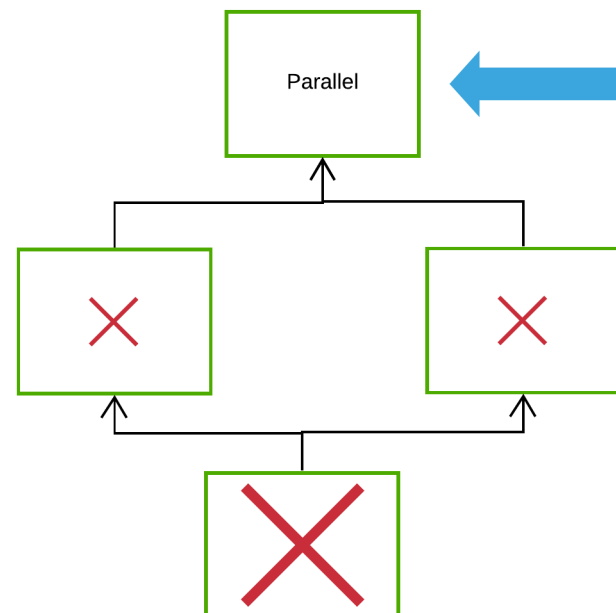
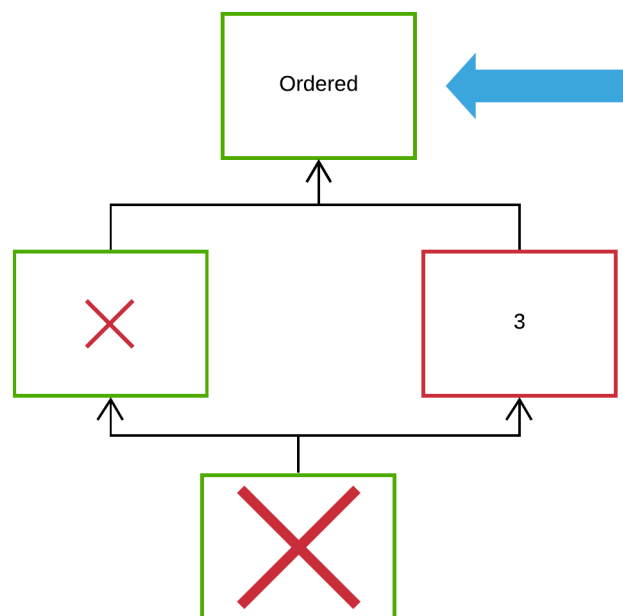
Workflow Failure



High Priority
Parallel
Ordered
Sequential
Success, Failure
Condition
Persist
Failed Items



Workflow Failure



High Priority

Parallel

Ordered

Sequential

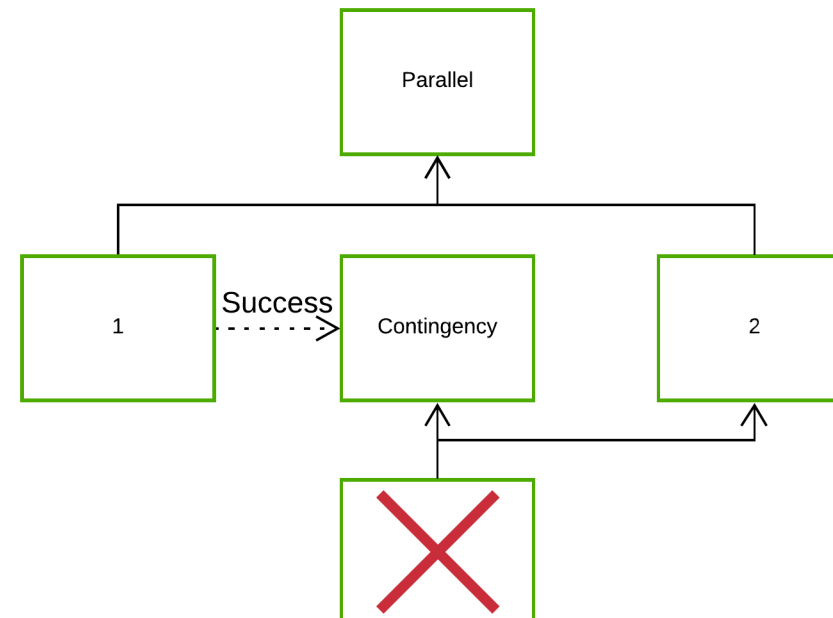
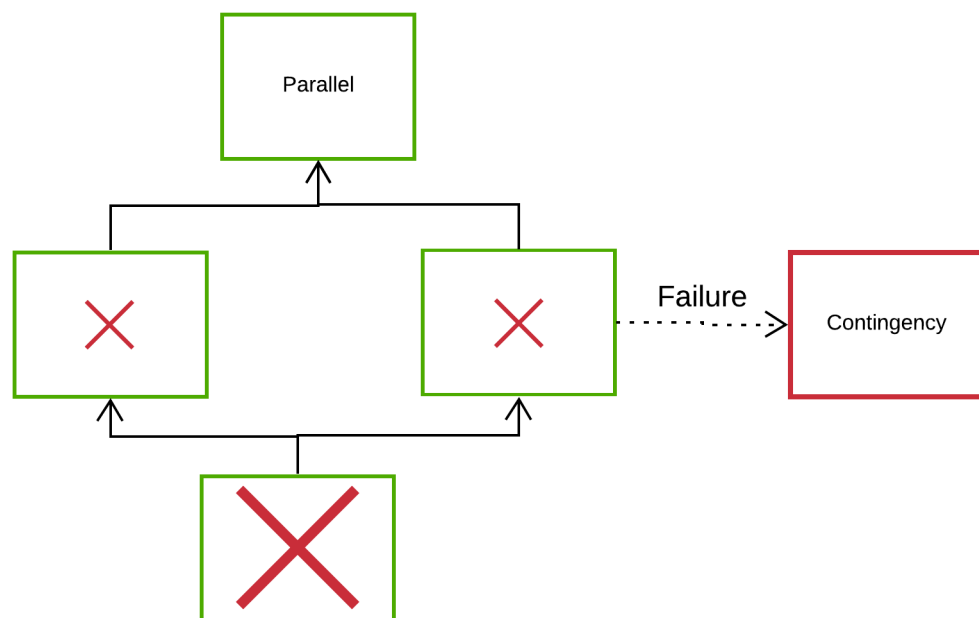
Success, Failure

Condition

Persist

Failed Items

Workflow Failure



High Priority

Parallel

Ordered

Sequential

Success, Failure

Condition

Persist

Failed Items

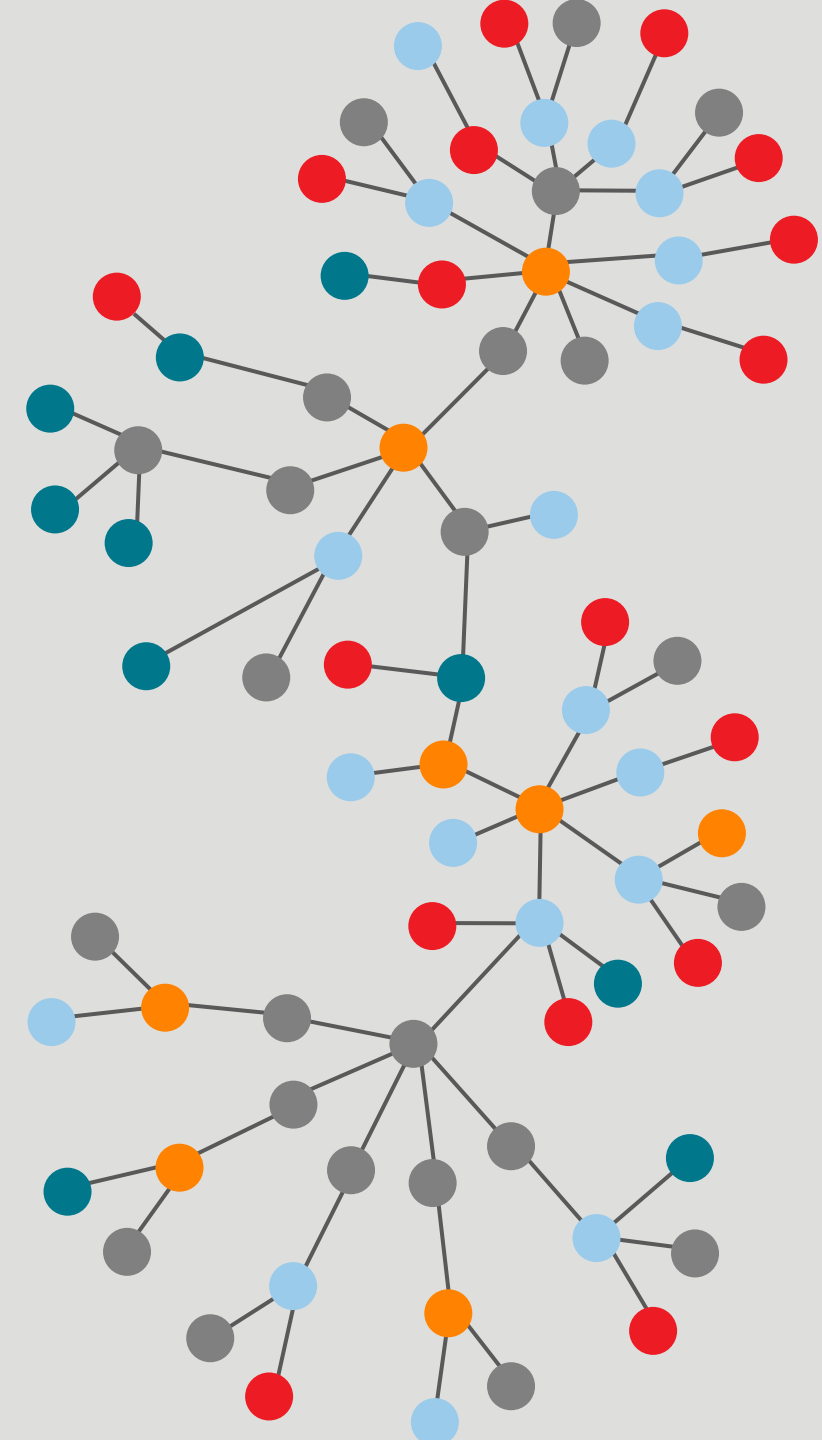
Benefits

- ✓ Faster workunits
- ✓ No time penalty for multi-purpose queries

Quick poll:

Will the parallel engine impact the way
that you write ECL queries?

See poll on bottom of presentation screen



Questions?

Nathan Halliday
High School Graduate



THE DOWNLOAD

TECH TALKS BY HPCC SYSTEMS

HPCC Systems on the Google Cloud Platform



Jeff Mao
High School Student
Lambert High School, Suwanee, GA

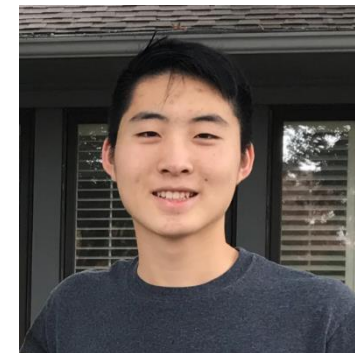


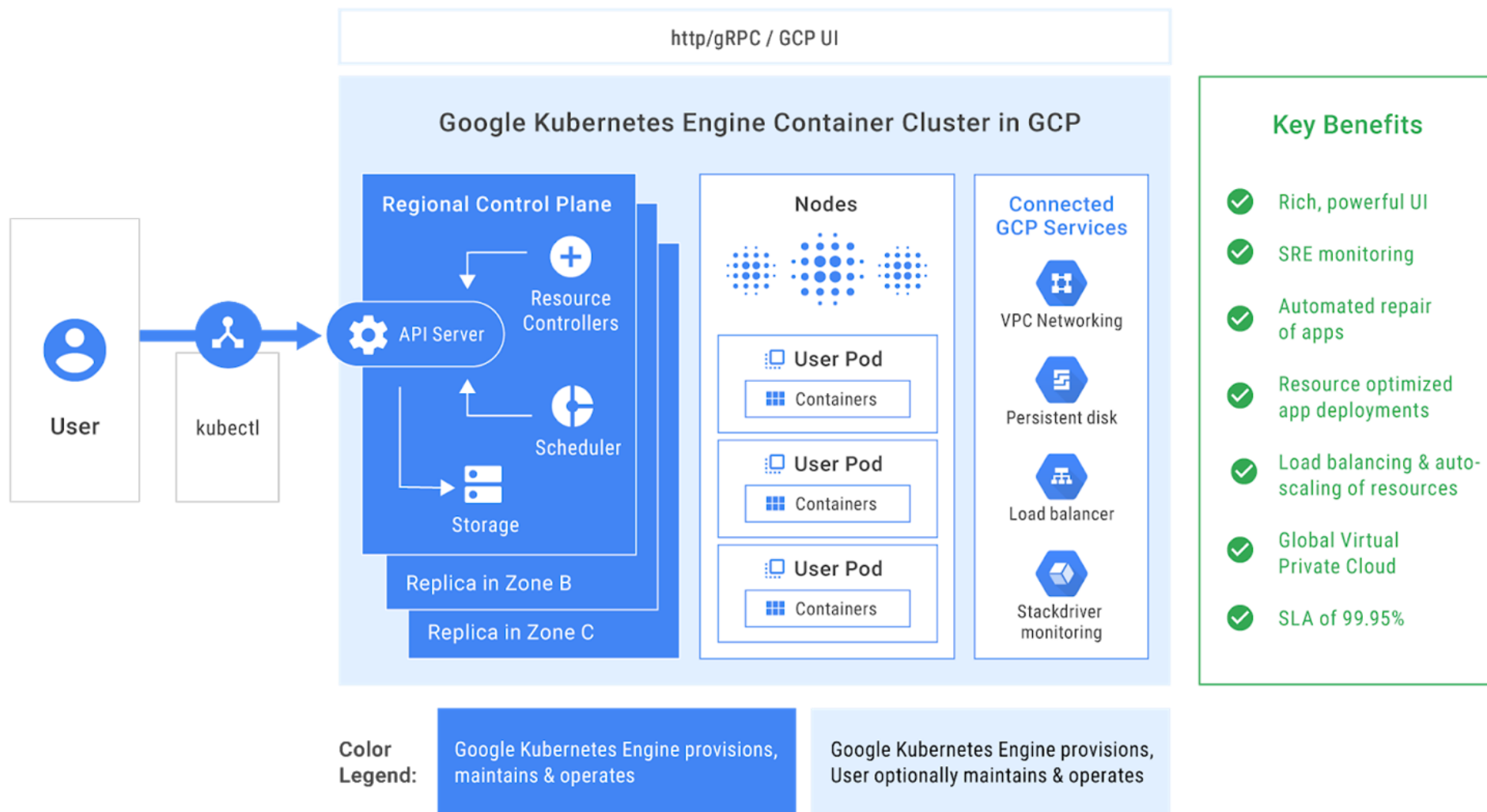
Table Of Contents

- Introduction to the Google Kubernetes Engine (GKE)
 - GKE Features
 - HPCC Systems Native Cloud Development
- HPCC Systems Cluster Deployment and Basic Tests
 - Regression Tests
 - ECL Playground
- GKE Autoscaling
 - Cluster Autoscaler
 - Horizontal/Vertical Autoscaler
- Introduction to Anthos
 - About Anthos
 - Anthos Components
 - Anthos Overview

GKE Features Overview



Google Kubernetes Engine Overview | **Highly Available with Regional Clusters**



HPCC Systems Native Cloud Development

Overview of HPCC Systems Native Cloud:

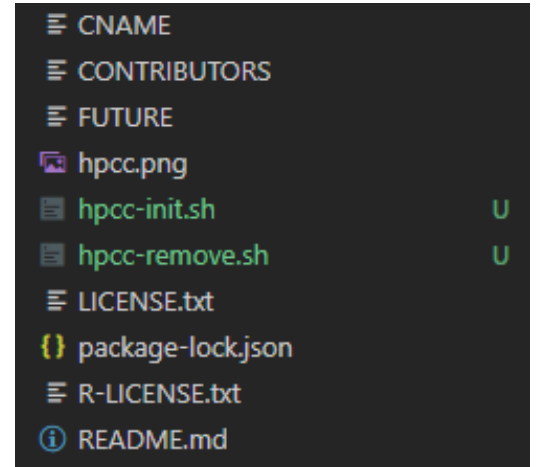
The HPCC Systems Native Cloud was implemented in version 7.8.0 of the HPCC Systems Platform. Therefore it is an expanding field, and not as well known as some of the other HPCC Systems Technology. Before, HPCC Systems had to be deployed via on premises resources through virtual machines. Now, deployment to the cloud is easier than ever. There are several blogs available on the HPCC Systems Wiki. <https://hpccsystems.com/blog>

Deployment to GKE:

The process of deploying HPCC Systems Native Cloud Development to the cloud is very simple.

I have created several scripts to streamline this process. It is located here:
<https://github.com/HypePhilosophy/hpcc-gcp>.

Clone hpcc-init and hpcc-remove shell scripts onto your project root path and run ./hpcc-init to deploy the project onto GKE or ./hpcc-remove to delete the existing project.



Quick poll:

Have you tried the Native HPCC Cloud Solutions?

See poll on bottom of presentation screen



Testing on the ECL Watch GKE Environment

ECL Playground Samples

- We tried a few samples from ECLWatch
- All samples were tested through the command line and received expected results

ECL Watch Data Tutorial

All tasks work on the HPCC Systems Data Tutorial with a few exceptions:

- Spraying and landing zone will not work (users will need to copy the data to the shared persistent volume)
- Publish Query through the browser is not supported

Testing Documentation: <https://hpccsystems.com/training/documentation/learning-ecl>

Regression Tests

- Development team is still working on regression testing. There are a few errors after running but that was the expected result

GKE Cluster Autoscaler

What does the Cluster Autoscaler do?

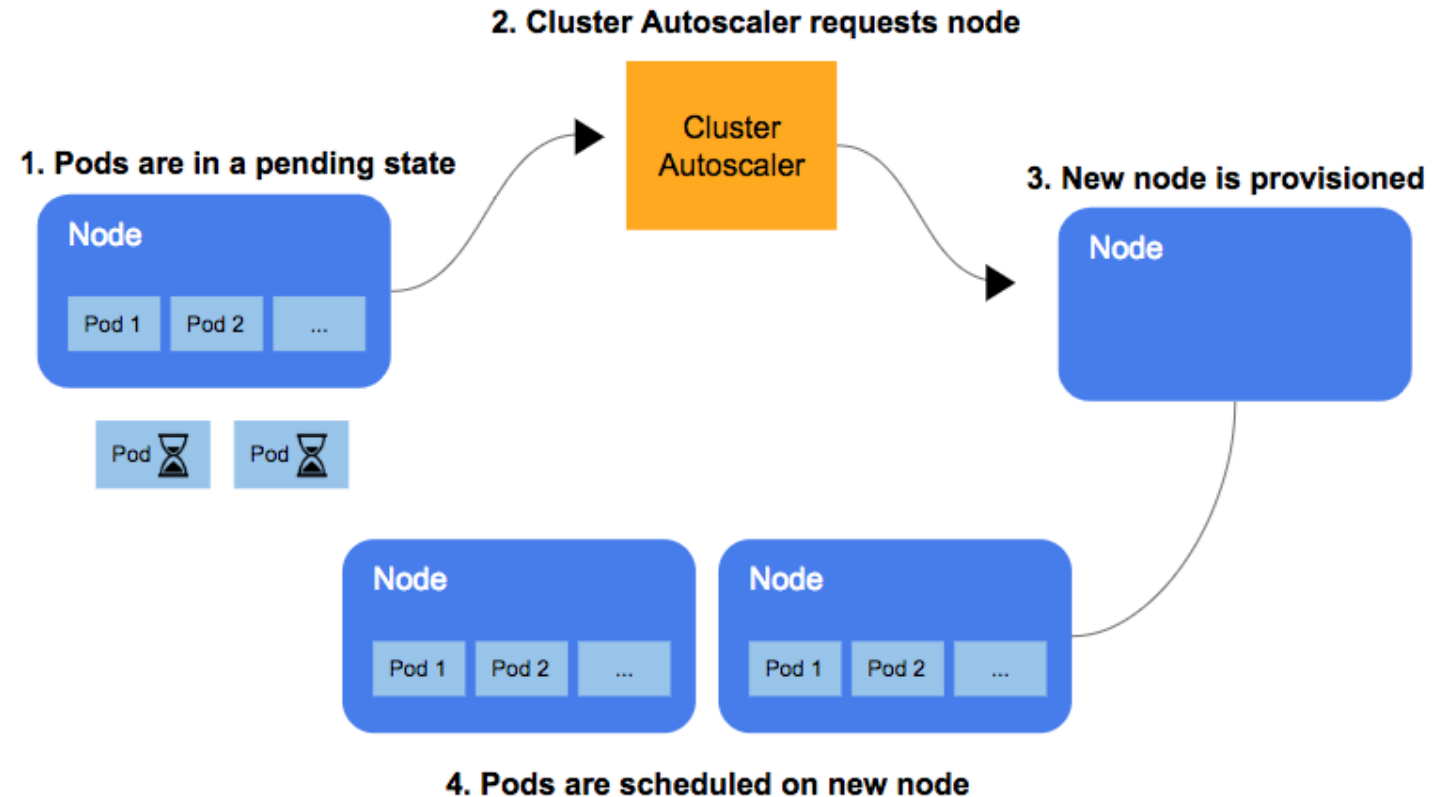
- The GKE Cluster Autoscaler automatically resizes the number of nodes per a given node pool.
- Threshold is set by Google K8s

How do you use it?

Enable Cluster Autoscaling and set the number of nodes per node pool.

Is it useful?

Cluster Autoscaler is very useful because instead of manually removing, adding, or over provisioning node pools, it performs this automatically, mitigating errors and cost/resource overruns.

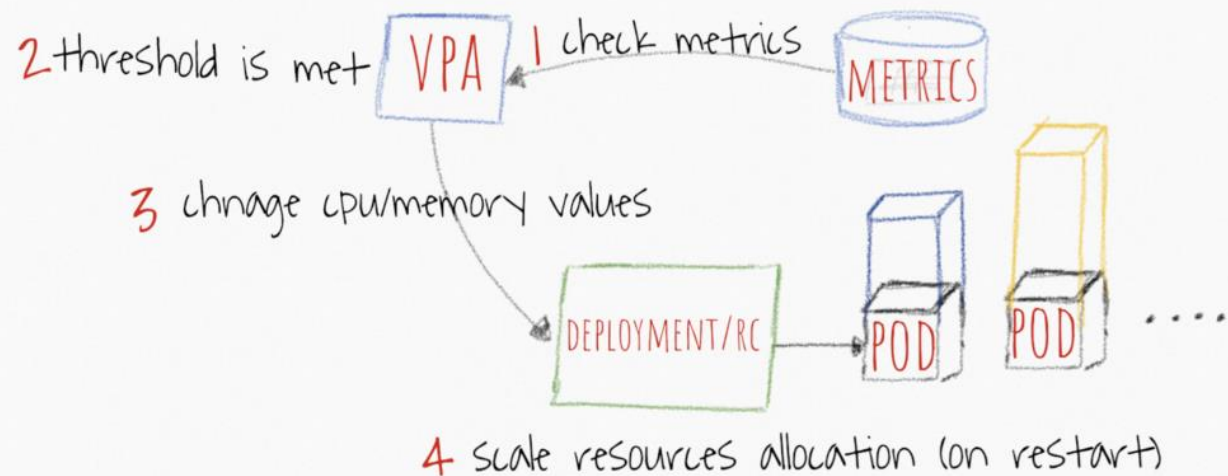
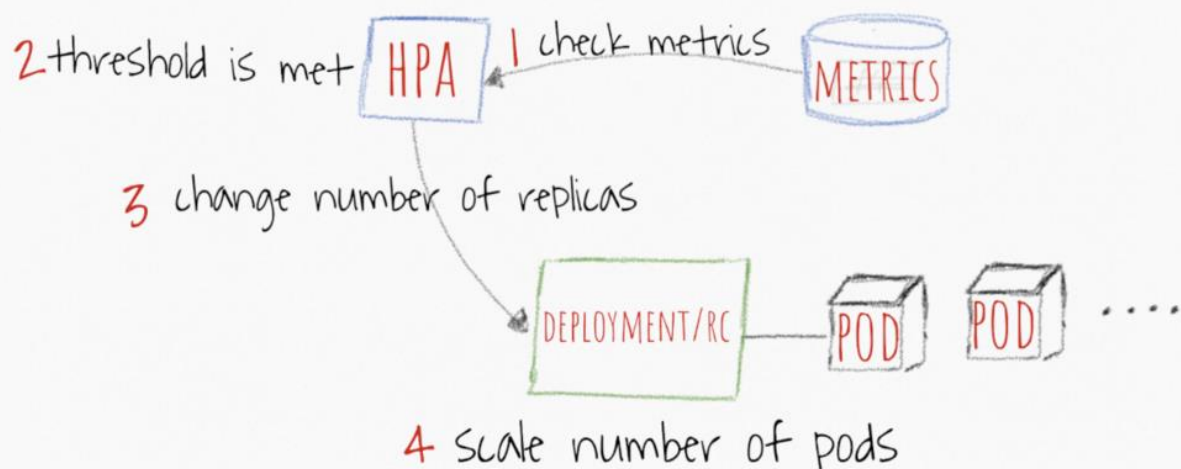


HPCC Pod Horizontal/Vertical Autoscaling

Horizontal Pod Autoscaler:

VS

Vertical Pod Autoscaler:



Anthos Defined

What is Google Anthos?

Anthos is a modern application management platform that provides a consistent development and operations experience for cloud and on-premises environments.

What does that mean?

It means that Google Anthos allows users to utilize other cloud resources such as Azure, Amazon, or even on Premise while taking full advantage of a singular advanced cloud environment.

Anthos Components:

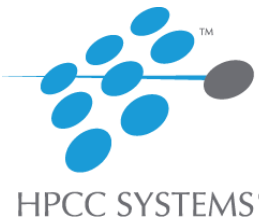
Anthos comes with a plethora of components to help users with application needs.

Anthos Components:

- Anthos GKE on-prem
- Google Kubernetes Engine
- Ingress for Anthos
- Migrate for Anthos
- Anthos Connect
- Config Management
- Service Mesh
- Logging and Monitoring
- Cloud Run for Anthos
- Binary Authorization
- Many more 3rd party Kubernetes apps



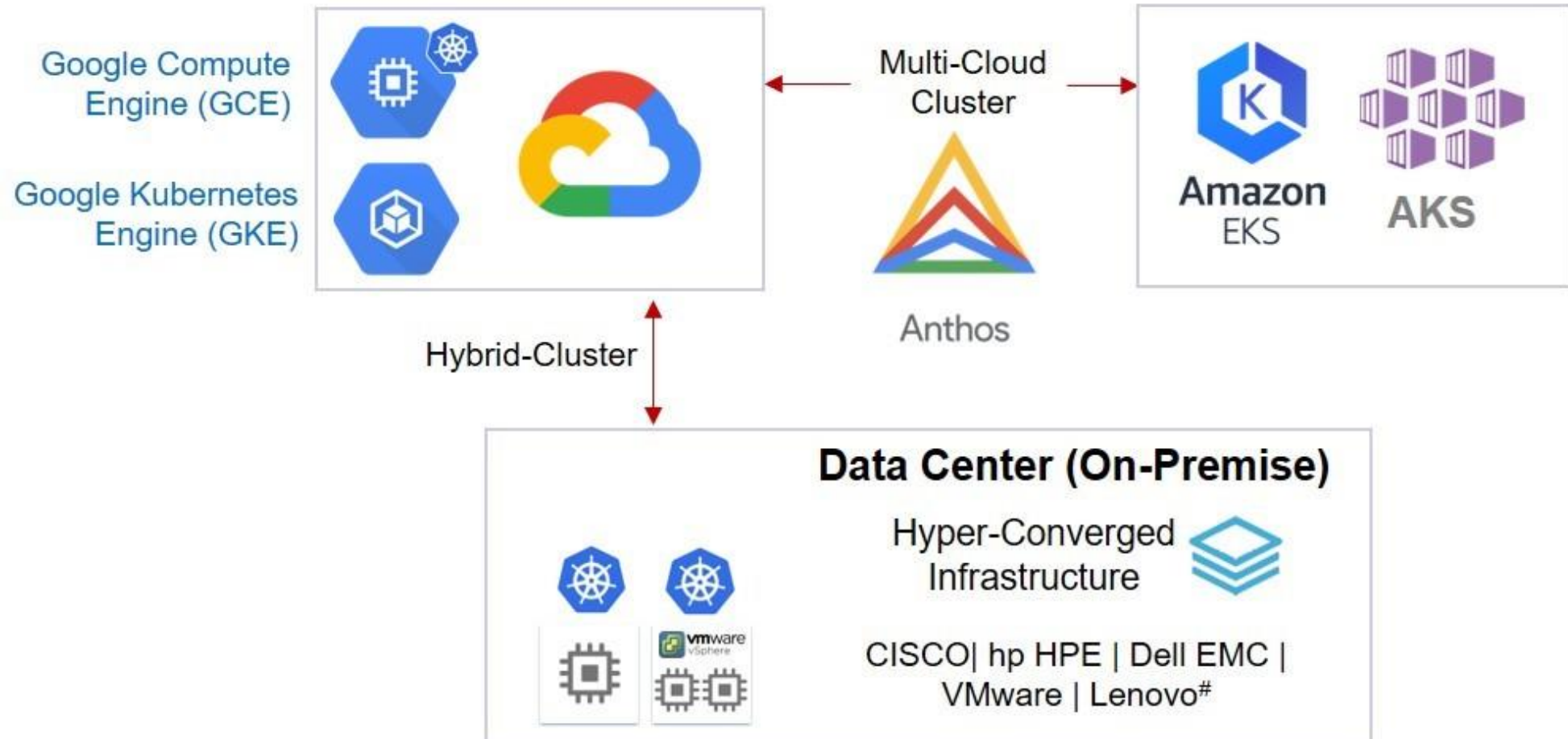
Anthos



Anthos Components Continued...

Core Anthos components	Cloud	On-premises	On Multi-Cloud
Infrastructure, container, and cluster management	Anthos GKE on Google Cloud Ingress for Anthos	Anthos GKE on-prem	Anthos GKE on AWS
Multicluster management	Anthos environ and Connect	Anthos environ and Connect	Anthos environ and Connect
Configuration management	Anthos Config Management Policy Controller Config Connector	Anthos Config Management Policy Controller Config Connector	Anthos Config Management Policy Controller
Migration	Migrate for Anthos		
Service management	Anthos Service Mesh (distribution) + Anthos Service Mesh dashboard + Anthos Service Mesh certificate authority	Anthos Service Mesh (distribution)	Anthos Service Mesh (distribution)
Serverless	Cloud Run for Anthos	Cloud Run for Anthos	
Secure software supply chain	Binary Authorization		
Logging and monitoring	Cloud Logging and Cloud Monitoring for system components	Cloud Logging and Cloud Monitoring for system components	
Marketplace	Kubernetes Applications in Cloud Marketplace	Kubernetes Applications in Cloud Marketplace	

Anthos Overview



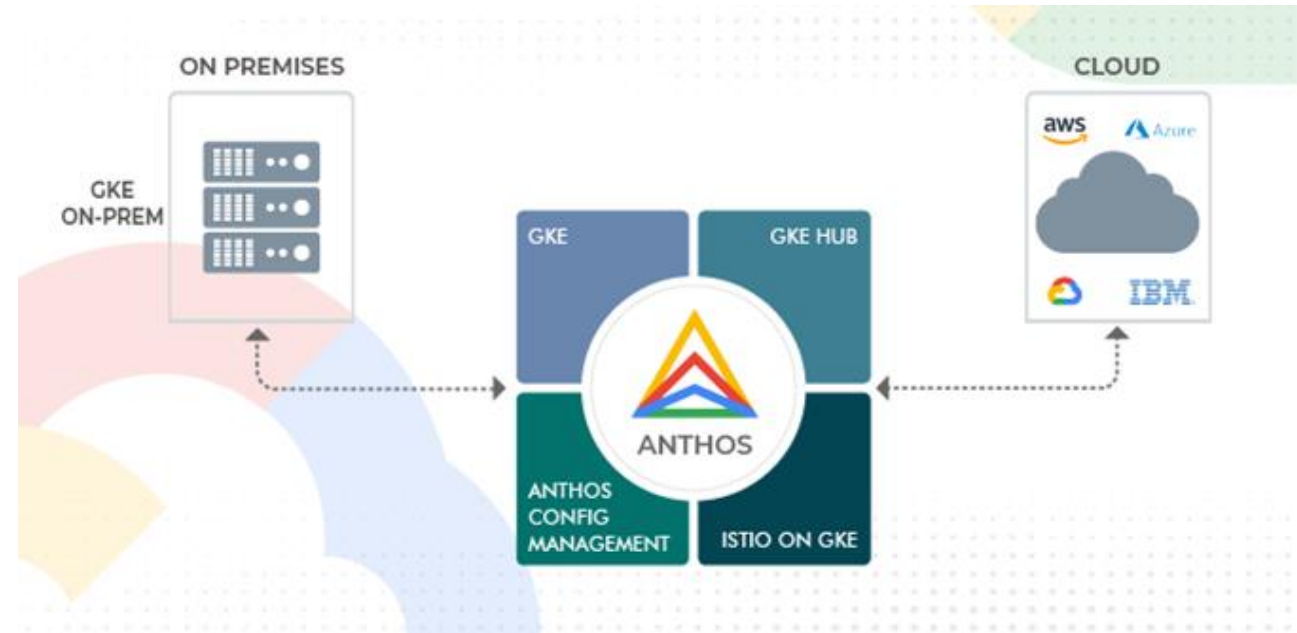
HPCC Systems Multi-Cloud Usage

HPCC Systems Multi-Cloud Examples:

1. HPCC Systems can function on multiple clouds in such a way that one cluster can be set for production, another for testing, and a final cluster for development.
2. On Azure, the majority of the HPCC Pods will function in one cluster, Thor pods in another cluster, and ECL Serviceless in the third.

How it works:

- Clusters will generally be in the same cloud environment.
- Clusters can also be separated into different clouds



Anthos Setup

Setup Scripts:

For the sake of simplicity and ease of use, several powershell scripts were created. These scripts are used to install Anthos components, HPCC Systems, and to register each cluster.

How to use them yourself:

Follow the readme in Anthos Setup Repository.

Script source: <https://github.com/HypePhilosophy/HPCC-Anthos-Setup>

```
✓ HPCC-ANTHOS-SETUP
  ✓ bin
    ➤ env.ps1 9+
    ➤ install-config-management.ps1
    ➤ install-hpcc.ps1
    ➤ install-istio.ps1
    ➤ register-clusters.ps1
    ➤ setup.ps1
    ➤ uninstall-hpcc.ps1
    ➤ unregister-clusters.ps1
```

Registered Anthos Clusters

Supported Clusters:

- EKS (Amazon)
- AKS (Azure)
- GKE (Google)

The screenshot displays the Anthos Clusters management interface. On the left is a navigation sidebar with options: Dashboard, Service Mesh, Config Management, Clusters (selected), Features, and Migrate to containers. The main content area is titled 'Clusters' with a 'BETA' badge and links for 'CREATE CLUSTER' and 'REGISTER EXISTING CLUSTER'. A 'Status' box indicates 'All clusters healthy' and '3 clusters total'. Below this is a table titled 'Anthos managed clusters' with a 'Filter table' option. The table lists three clusters: 'aks-myhpcc-admin' (azure-aks, External), 'eks-hpcc-1' (registered, External), and 'gke-hpcc-1' (us-east1-b, GKE). Each row has a green checkmark icon in the first column.

	Name ↑	Location	Type	Labels
✓	aks-myhpcc-admin	azure-aks	External	location : azure-aks mesh_id : proj-193926534835
✓	eks-hpcc-1	registered	External	
✓	gke-hpcc-1	us-east1-b	GKE	

In the image above you can see that all registered clusters can be managed through Anthos. This gives users a plethora of options including but not limited to: Ingress, Config Management, Service-Mesh, Cloud Run, etc.

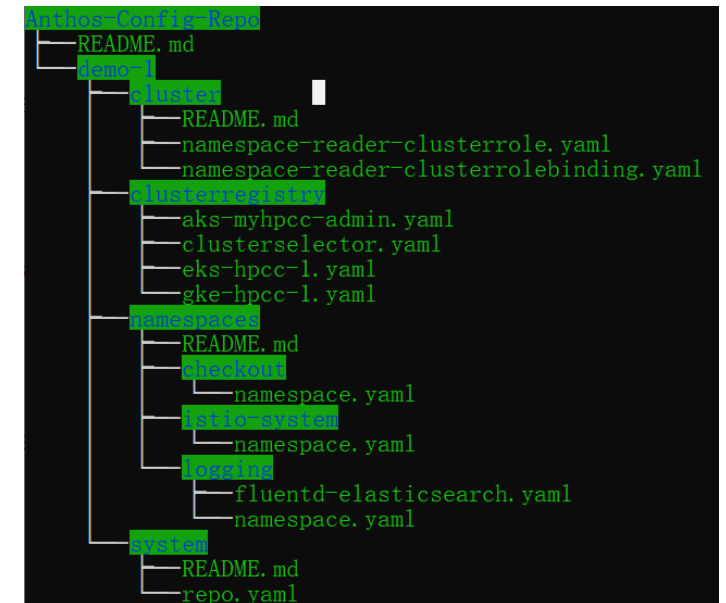
Config Management

What is Config Management?

Anthos Config Management is a service which allows you to create a common configuration across all your infrastructure, including custom policies, and apply it both on-premises and across clouds.

Default Repository Structure:

- | | |
|-------------------|---|
| - cluster | - [Required] Contains configs that pertain to entire clusters. |
| - clusterregistry | - [Optional] Contains configs for ClusterSelectors . ClusterSelectors limit which clusters a config applies to. |
| - namespaces | - [Required] Contains configs for namespaces and namespace-scoped objects. |
| - system | - [Required] This directory contains configs for the operator. |



Anthos Config Management Dashboard

```
apiVersion: configmanagement.gke.io/v1
kind: ConfigManagement
metadata:
  name: config-management
  namespace: config-management-system
spec:
  #clusterName: <CLUSTER-NAME>
  git:
    syncRepo: https://github.com/HypePhilosophy/HPCC-Anthos-Setup
    syncBranch: master
    secretType: none
    policyDir: "demo-1"
    syncWait: 2
  enableAggregateNamespaceQuotas: true
  enableHierarchicalResourceQuota: true
```

- Anthos Config Management Dashboard is set up using the config-management.yaml file. This tells Anthos which repository, branch, and policy directory to follow.
- From there, users can access the dashboard to check basic information like status, version, latest commit, etc

Anthos Config Management ⚙️ CONFIGURE

Clusters for "anthos-hpcc-1"

Filter table

<input type="checkbox"/>	Name ↑	Status	Last Synced	Sync Branch	Sync Tag	Commit
<input type="checkbox"/>	aks-myhpcc-admin	✓ Synced	5 days ago	master		47322229adbd80a6bc8b928c323d3c11862b26fb
<input type="checkbox"/>	eks-hpcc-1	✓ Synced	5 days ago	master		47322229adbd80a6bc8b928c323d3c11862b26fb
<input type="checkbox"/>	gke-hpcc-1	✓ Synced	2 days ago	master		47322229adbd80a6bc8b928c323d3c11862b26fb

All clusters are synced despite being provided from different cloud providers.

Anthos ← Clusters 🔧 UPDATE CLUSTER

☒ eks-hpcc-1

DETAILS

Basic information

Cluster name	eks-hpcc-1
Status	Synced
Sync repo	https://github.com/xwang2713/Anthos-Config-Repo
Sync branch	master
Last sync	5 days ago
Secret type	none
Policy directory	demo-1
Sync wait ?	2
Sync tag ?	

How to Utilize Anthos Config Management

For this example we will be creating a namespace. We will also create a fluentd-elasticsearch pod for all clusters.

```
$ cat namespace.yaml
apiVersion: v1
kind: Namespace
metadata:
  name: checkout
  annotations:
    configmanagement.gke.io/cluster-selector: selector-env-aks
```

```
$ cat clusterselector.yaml
kind: ClusterSelector
apiVersion: configmanagement.gke.io/v1
metadata:
  name: selector-env-aks
spec:
  selector:
    matchLabels:
      env: aks-myhpcc-admin
      lifecycle: prod
```

1. Create the checkout namespace if it does not exist.
2. Create the clusterselector for the checkout namespace and utilize the selector from the namespace.yaml file.

Config Management Code:

<https://github.com/HypePhilosophy/hpcc-anthos>

This config file installs the daemonset image fluentd for all the managed Kubernetes clusters

```
$ cat fluentd-elasticsearch.yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: fluentd-elasticsearch
  namespace: logging
  labels:
    k8s-app: fluentd-logging
spec:
  selector:
    matchLabels:
      name: fluentd-elasticsearch
  template:
    metadata:
      labels:
        name: fluentd-elasticsearch
    spec:
      tolerations:
        - key: node-role.kubernetes.io/master
          effect: NoSchedule
      containers:
        - name: fluentd-elasticsearch
          image: quay.io/fluentd_elasticsearch/fluentd:latest
```

Managed cluster

Everytime
Kubernetes Nodes
scales up, each new
node will contain a
fluentd-elasticsearch
pod

What is Istio Service Mesh?

- What is a Service Mesh?

A service mesh is an infrastructure layer that enables managed, observable, and secure communication across your services, letting you create robust enterprise applications made up of many microservices on your chosen infrastructure.

- What is Istio?

Istio is a powerful and highly configurable open source Service Mesh platform which features many tools and services that enables applications to follow industry best practices.

- Installation:

Run the `./install-istio.ps1` script from the HPCC-Anthos-Setup Github.

<https://github.com/HypePhilosophy/HPCC-Anthos-Setup>

Istio and HPCC Systems Cluster Install

Istio Installed

```
(base) PS C:\> kubectl get pods -n istio-system
```

NAME	READY	STATUS	RESTARTS	AGE
grafana-5dc4b4676c-xf2tj	1/1	Running	0	5d
istio-ingressgateway-749d6659ff-dqv22	1/1	Running	0	5d
istio-ingressgateway-749d6659ff-s2t28	1/1	Running	0	5d
istiod-69657b479-99x5r	1/1	Running	0	5d
istiod-69657b479-f2v7m	1/1	Running	0	5d
kiali-6f457f5964-t7fbz	1/1	Running	0	5d
prometheus-6b567696c5-wjp6h	2/2	Running	0	5d
promsd-6b77b75f8b-2d46j	2/2	Running	1	5d

```
(base) PS C:\> █
```

HPCC Systems Installed

```
(base) PS C:\> kubectl get pods -n hpcc-system
```

NAME	READY	STATUS	RESTARTS	AGE
eclqueries-784dbc977c-th4kr	2/2	Running	0	26h
eclservices-6859f8d59c-75gsz	2/2	Running	0	26h
eclwatch-cff54f7bd-8v1xc	2/2	Running	0	26h
efs-provisioner-efsstorage-hpcc-efs-b4485996c-n4kkf	2/2	Running	2	26h
hthor-5b5c957896-8z8lt	2/2	Running	0	26h
mydali-557ff6c7f4-fjw88	2/2	Running	0	26h
myeclccserver-b9b969588-76w62	2/2	Running	0	26h
roxie-6d8579d6d8-x9bg6	2/2	Running	0	26h
roxie-cluster-slave-1-7f6dbc89df-8jk4p	2/2	Running	0	26h
roxie-cluster-slave-2-cddff4664-dl2c6	2/2	Running	0	26h
roxie-cluster-toposerver-65df8dc9ff-wd7j2	2/2	Running	0	26h
sql2ecl-69cdb475cf-fdt8c	2/2	Running	0	26h
thor-agent-548b87bb77-csmzt	2/2	Running	0	26h
thor-thoragent-574db4457d-psqkz	2/2	Running	0	26h

Check out JIRA HPCC-24548 for more information



Ingress Gateway Configuration

What does Ingress Gateway do?

Istio Ingress Gateway is a gateway network management system that defines rules for routing external HTTP/TCP traffic to services in a Kubernetes cluster.

Example Usage:

Routing 20% of requests to a new version while keeping 80% of requests on stable version.

Why use Ingress Gateway?

It's fast, secure, and easy to implement.

This is the ecl-gateway.yaml file used to configure istio ingressgateway to route ECLWatch requests

```
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
  name: eclwatch-gateway
  namespace: istio-system
spec:
  selector:
    istio: ingressgateway # use istio default controller
  servers:
  - port:
      number: 8010
      name: http
      protocol: HTTP
    hosts:
    - "*"

---
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: eclwatch
  namespace: istio-system
spec:
  hosts:
  - "*"
  gateways:
  - eclwatch-gateway
  http:
  - match:
    - port: 8010
      uri:
        prefix: /
    route:
    - destination:
        host: eclwatch.hpcc-system.svc.cluster.local
```

All requests to ECLWatch will be routed through Istio Ingress Gateway.

Logging on the Google Cloud Platform

There are a wide variety of native in house Google K8s logging tools available to users. Anthos adds upon this by allowing more detailed 3rd party metrics and graphs to be used.

If third party solutions are not enough, users can also opt to create their own graphs and charts through Google's intuitive console UI.

Services Overview + DEFINE SERVICE AUTO REFRESH

We currently support four service types for auto-ingestion: [Anthos Service Mesh](#), [Istio on GKE](#), [App Engine](#), and [Endpoints](#). We also support custom [user-defined microservices](#). Microservices are a monitoring construct to enable better observability. No code changes are required to enable this feature. [Learn more](#)

DISMISS DEFINE SERVICE

Current status of 10 services Status was calculated at 4:40 PM

SLO alert firing -

SLO out of budget -

Continue service setup

10 services with no SLOs set [Show Services](#)

Define Kubernetes Engine service [Define service](#)

Name ↑	Type	SLOs out of error budget	SLOs with firing alert	Labels
ecqueries	Anthos	0 / 0	0 / 0	project_id: intern-projects-278514 mesh_uid: proj-616121786494 service_namespace: default service_name: ecqueries
ecservices	Anthos	0 / 0	0 / 0	project_id: intern-projects-278514 mesh_uid: proj-616121786494 service_namespace: default service_name: ecservices
ecwatch	Custom	0 / 0	0 / 0	GKE: Deployment project_id: intern-projects-278514 location: us-east1-b cluster: gke-hpcc-3 namespace: default deployment: ecwatch
ecwatch	Anthos	0 / 0	0 / 0	project_id: intern-projects-278514 mesh_uid: proj-616121786494 service_namespace: default service_name: ecwatch
istio-ingressgateway	Anthos	0 / 0	0 / 0	project_id: intern-projects-278514 mesh_uid: proj-616121786494 service_namespace: istio-system service_name: istio-ingressgateway
mydali	Anthos	0 / 0	0 / 0	project_id: intern-projects-278514 mesh_uid: proj-616121786494 service_namespace: default service_name: mydali
nfsstorage-hpcc-nfs	Anthos	0 / 0	0 / 0	project_id: intern-projects-278514 mesh_uid: proj-616121786494 service_namespace: default service_name: nfsstorage-hpcc-nfs
roxie-cluster-query	Anthos	0 / 0	0 / 0	project_id: intern-projects-278514 mesh_uid: proj-616121786494 service_namespace: default service_name: roxie-cluster-query
roxie-cluster-toposerver	Anthos	0 / 0	0 / 0	project_id: intern-projects-278514 mesh_uid: proj-616121786494 service_namespace: default service_name: roxie-cluster-toposerver
sql2ec1	Anthos	0 / 0	0 / 0	project_id: intern-projects-278514 mesh_uid: proj-616121786494 service_namespace: default service_name: sql2ec1

Shown above is the GKE Logging and Monitoring Dashboard. This provides very detailed information on specific services inside GKE. Each service can be expanded to show logs, graphs, and errors. This only works on GKE clusters. There are other Anthos Logging and Monitoring Solutions that specifically serve multi-cloud purposes.

Monitoring GKE Clusters with Anthos

Define a custom service

Define a custom service by selecting a GKE entity from the list below. Services are a monitoring construct to enable better observability. No code changes are required to enable this feature.

Select a GKE entity

Filter table					
Name	Type	Labels			
thor-thoragent	Deployment	project_id: intern-projects-278514	location: us-east1-b	cluster: gke-hpcc-3	▼
ecqueries	Kubernetes Service	project_id: intern-projects-278514	location: us-east1-b	cluster: gke-hpcc-3	▼
ecservices	Kubernetes Service	project_id: intern-projects-278514	location: us-east1-b	cluster: gke-hpcc-3	▼
ecwatch	Kubernetes Service	project_id: intern-projects-278514	location: us-east1-b	cluster: gke-hpcc-3	▼
kubernetes	Kubernetes Service	project_id: intern-projects-278514	location: us-east1-b	cluster: gke-hpcc-3	▼
mydali	Kubernetes Service	project_id: intern-projects-278514	location: us-east1-b	cluster: gke-hpcc-3	▼
nfsstorage-hpcc-nfs	Kubernetes Service	project_id: intern-projects-278514	location: us-east1-b	cluster: gke-hpcc-3	▼
roxie-cluster-query	Kubernetes Service	project_id: intern-projects-278514	location: us-east1-b	cluster: gke-hpcc-3	▼
roxie-cluster-toposerver	Kubernetes Service	project_id: intern-projects-278514	location: us-east1-b	cluster: gke-hpcc-3	▼
sql2ec1	Kubernetes Service	project_id: intern-projects-278514	location: us-east1-b	cluster: gke-hpcc-3	▼

Rows per page: 10 21 - 30 of 68

Service details

Display name
ecservices

SUBMIT CLOSE

Anthos Service Mesh

Namespace: Any namespace ☐ Show system services Time Span: 1 hour

Current status of 8 services
Status was calculated at 11:22 PM

No alerts firing 0 Filter by	Alerts firing 0 Filter by	SLOs out of error budget 0 Filter by	No SLO alerting policies 0 Filter by	No SLOs set 8 Filter by
--	---	--	--	---

Services

Status	Name	Namespace	Type	Clusters	Requests/sec (avg)	Error rate	50% latency	99% latency
●	ecqueries	default	K8s	gke-hpcc-1	-	-	-	-
●	ecservices	default	K8s	gke-hpcc-1	-	-	-	-
●	ecwatch	default	K8s	gke-hpcc-1	-	-	-	-
●	mydali	default	K8s	gke-hpcc-1	-	-	-	-
●	nfsstorage-hpcc-nfs	default	K8s	gke-hpcc-1	-	-	-	-
●	roxie-cluster-query	default	K8s	gke-hpcc-1	-	-	-	-
●	roxie-cluster-toposerver	default	K8s	gke-hpcc-1	-	-	-	-
●	sql2ec1	default	K8s	gke-hpcc-1	-	-	-	-

1. From the Google Cloud Console, select “Monitoring/Service”
2. Select “Define Service”
3. Select any entry out of the list
4. Submit the service
5. That service will now appear in your Anthos Service Mesh Dashboard!
6. From the Anthos Mesh Dashboard, you are able to set service-level objectives for your services!

Monitoring Non-GKE Clusters with Anthos

Setup:

Third party services such as Grafana and Kiali to monitor Non-GKE clusters require users to set up each service. You also have to apply the VirtualServices for each third party solution.

Most third party graphing solutions utilize **Prometheus** metrics

What is Prometheus?

Prometheus is a monitoring tool often used with Kubernetes. It can be toggled on and off and will show detailed information on your clusters.

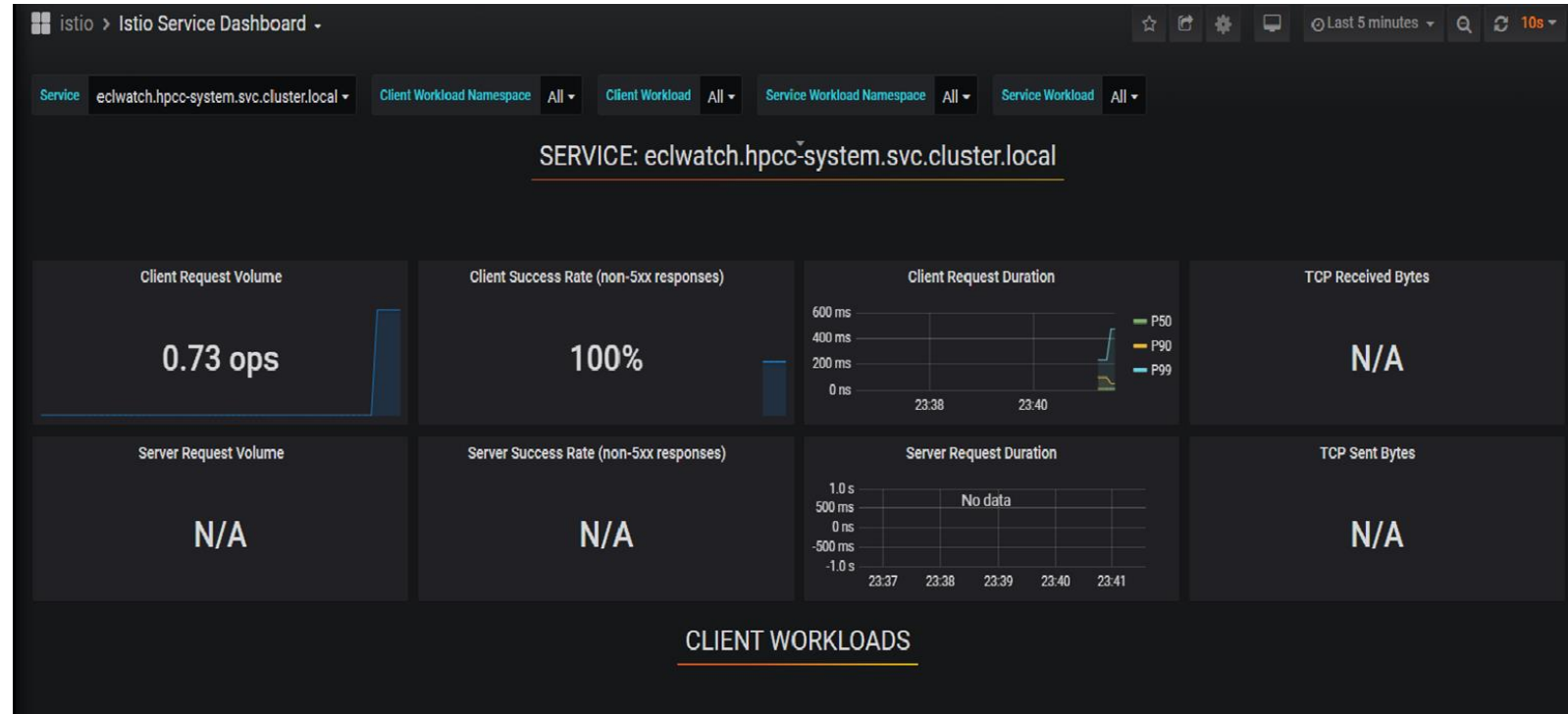
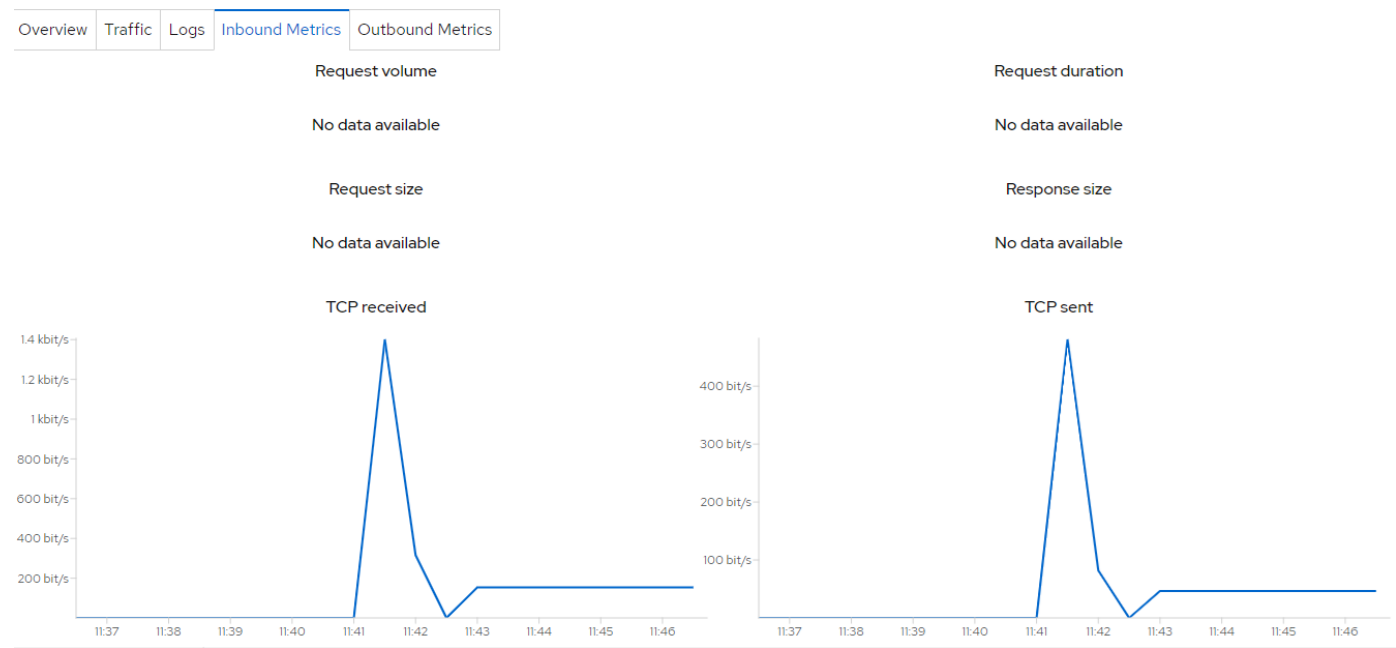


Image from HPCC Systems Grafana Dashboard

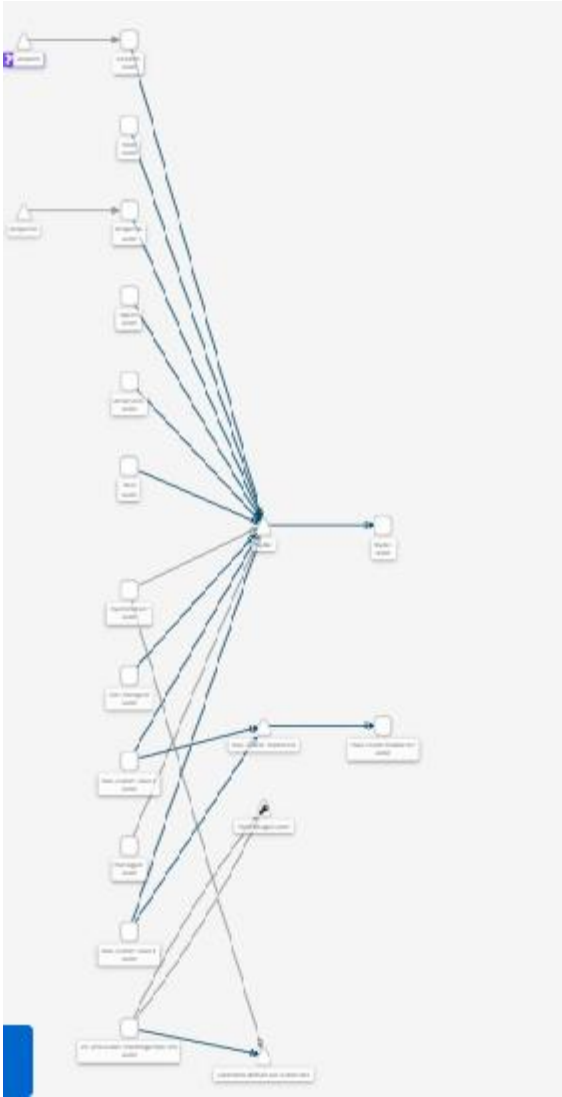
Monitoring Non-GKE Clusters with Anthos Continued...

Kiali

ECL Watch requests:



HPCC Systems Graph:



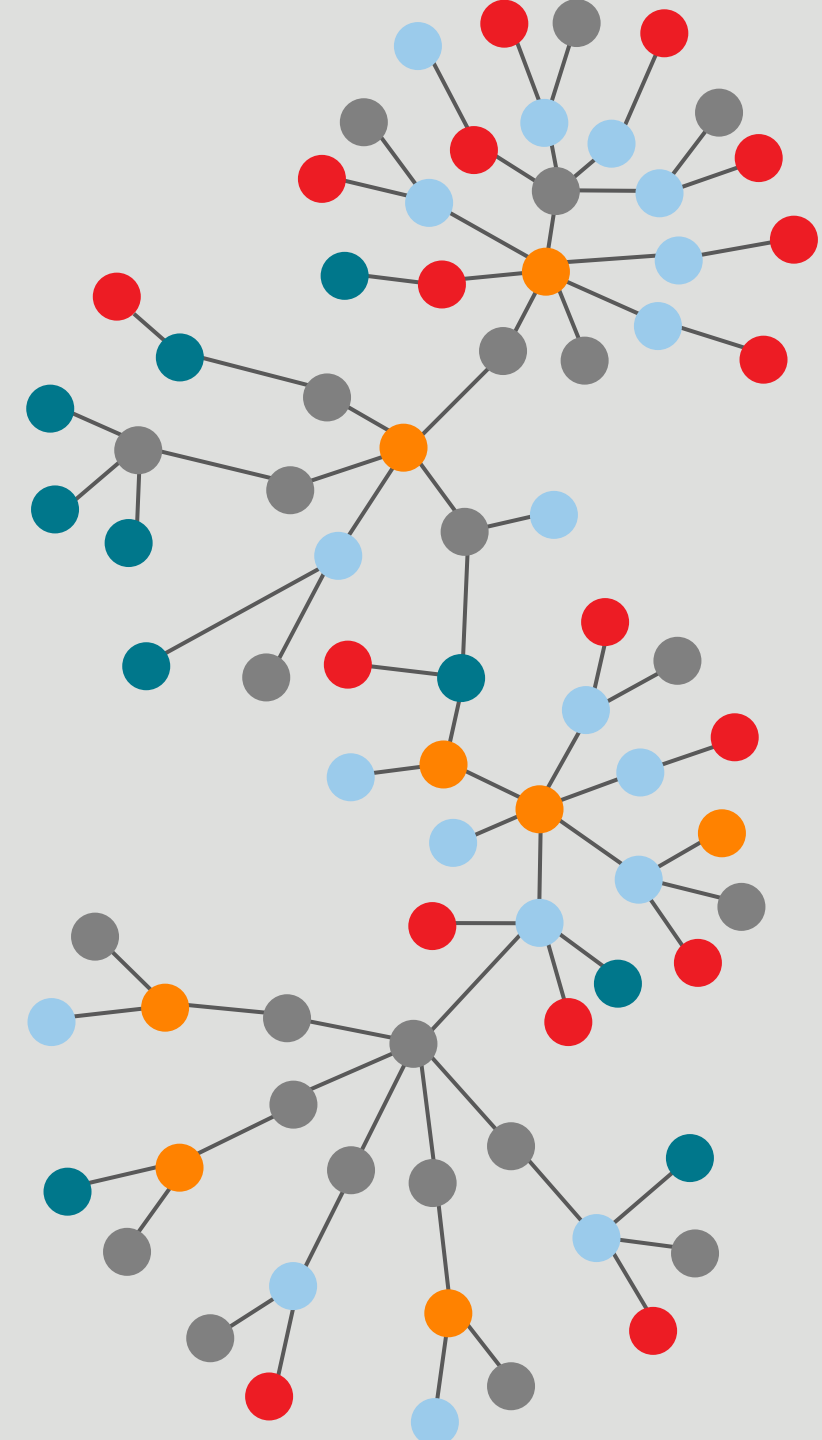
Takeaway

- Anthos is a trend for multi-cloud kubernetes management platform.
- Useful Components (Can apply to individual k8s clusters, not just anthos)
 - Configure Management
 - Istio Envoy, Monitoring & Logging, etc
- Anthos is still under development, i.e. the current lack of support for GKE on Azure.
- Since Kubernetes Cluster Management is important, we may evaluate and compare other technologies, such as Azure Stack/Arc, AWS Outposts and Rancher, etc
- The Google Cloud Platform is very versatile and is always expanding

Quick poll:

Do you use Kubernetes Management Systems in your work?

See poll on bottom of presentation screen



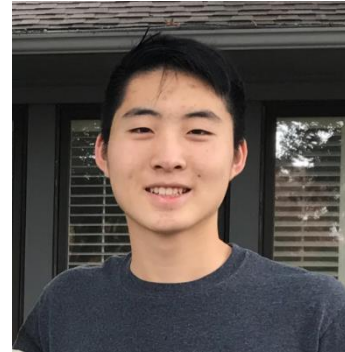
Useful Links

- <https://github.com/HypePhilosophy/hpcc-gcp>
HPCC Systems Setup Scripts
- <https://github.com/HypePhilosophy/HPCC-Anthos-Setup>
Anthos Component Setup Scripts
- <https://github.com/HypePhilosophy/hpcc-anthos>
Anthos Config Management Files
- <https://hpccsystems.com/blog>
HPCC Systems blog

Questions?

Jefferson “Jeff” Mao

High School Student
Lambert High School, Suwanee, GA
jeffs.mao@gmail.com



THE DOWNLOAD

TECH TALKS BY HPCC SYSTEMS



Continuing the integration with Robotics and HPCC SYSTEMS and using the GNN Bundle with TensorFlow

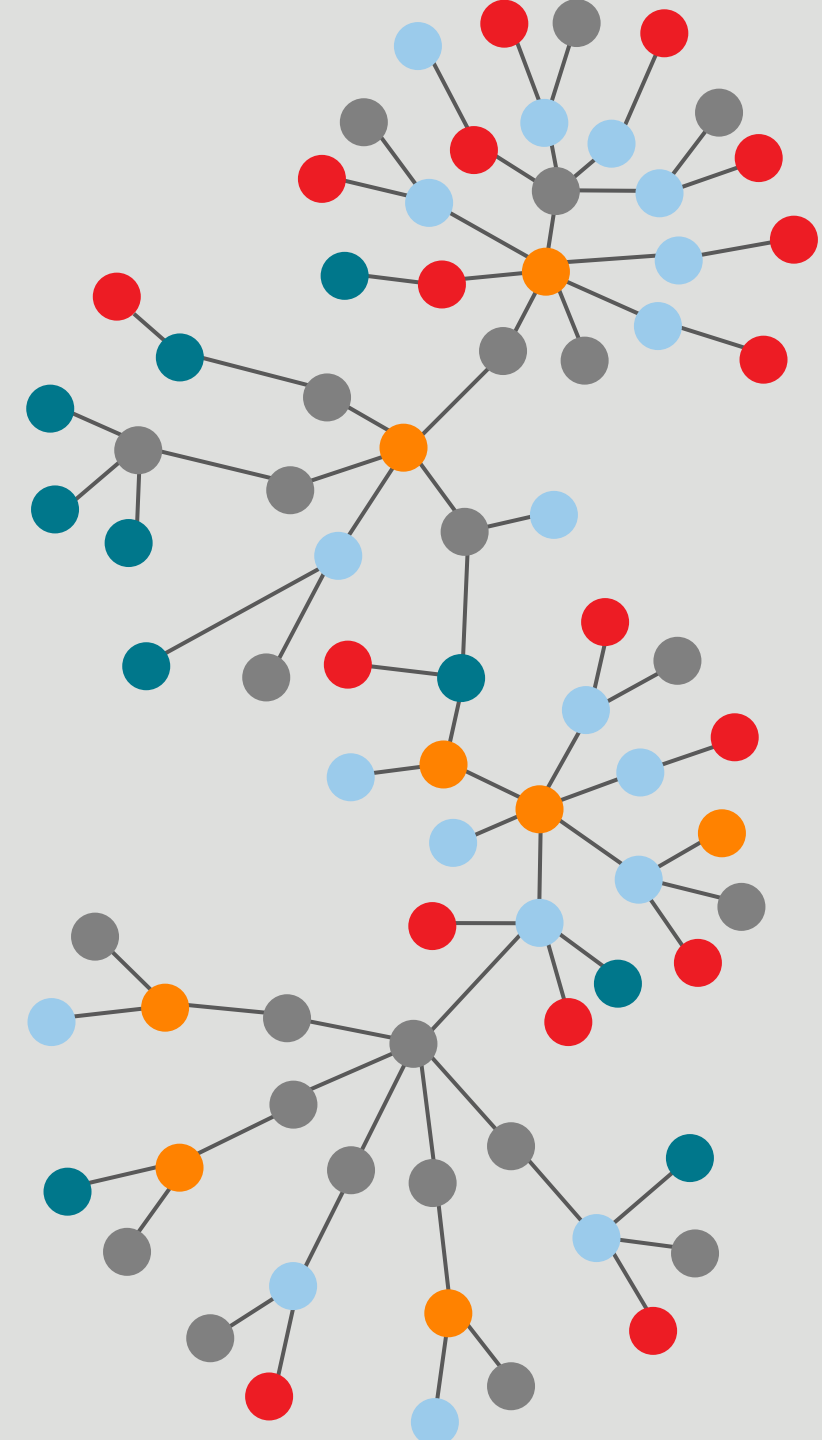
Jack Fields
High School Student
American Heritage School, Delray Beach, FL



Quick poll:

Have you ever used ROS?

See poll on bottom of presentation screen



What is ROS?

 ROS

Background

Previous intern from American Heritage Aramis Tanelus created an API for ingesting DATA to HPCC SYSTEMS.

The software for this project is currently being implemented into the American Heritage Security Robot seem at the 2019 Summit



American Heritage Security Robot



- Use Robotics to make school campuses safer
- Autonomous Driving
- Collect data to spray into HPCC SYSTEMS

Project Focus

The Focus of this project is not only ingest data from sensors but to process it.

The goal of this project was to also processes data that would happen in a real world scenario.

Post Project

Updating actual Robot code to newest version. Fixing Command base structure.

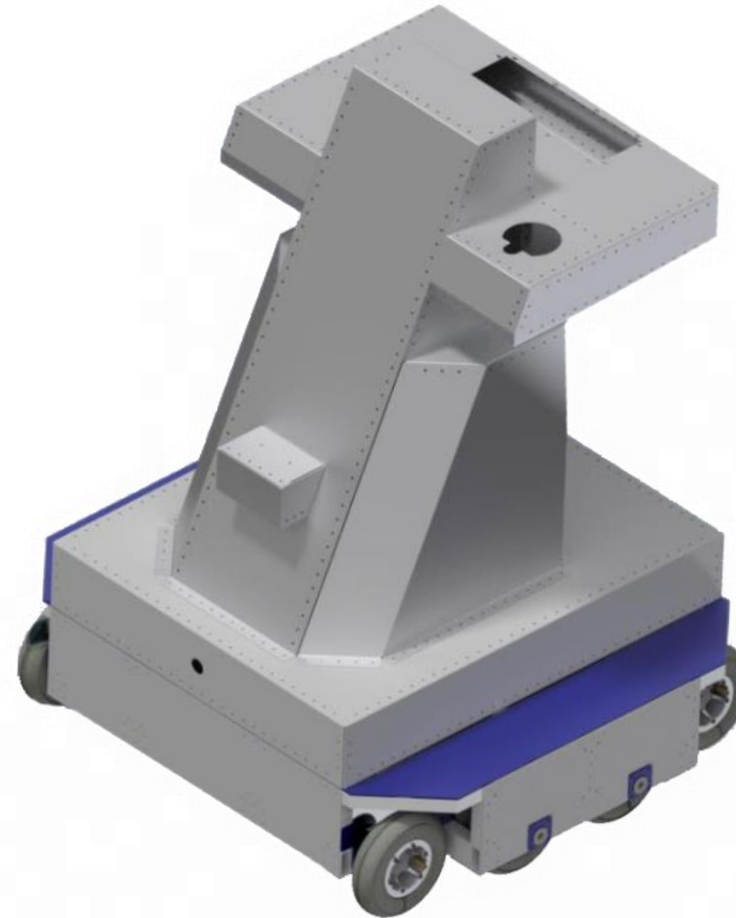
Updating to the Newest Version of ROS



What Kind of DATA are we looking at?

Sensor Data

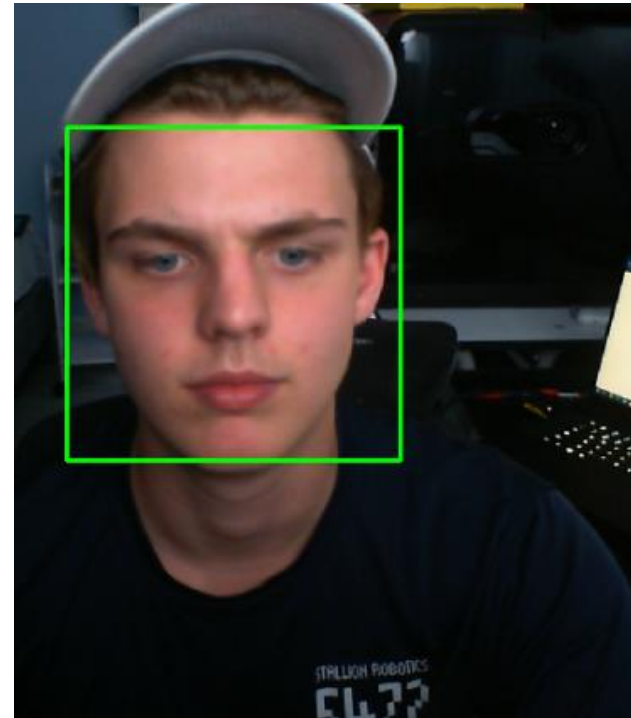
- Lidar
- Zed Stereo Cameras
- RGB Cameras
- GNSS and RTK positioning
- Scanning Rangefinder
- Infrared Cameras



Data Collection

Develop Software to allow a person to scan themselves into the robot.

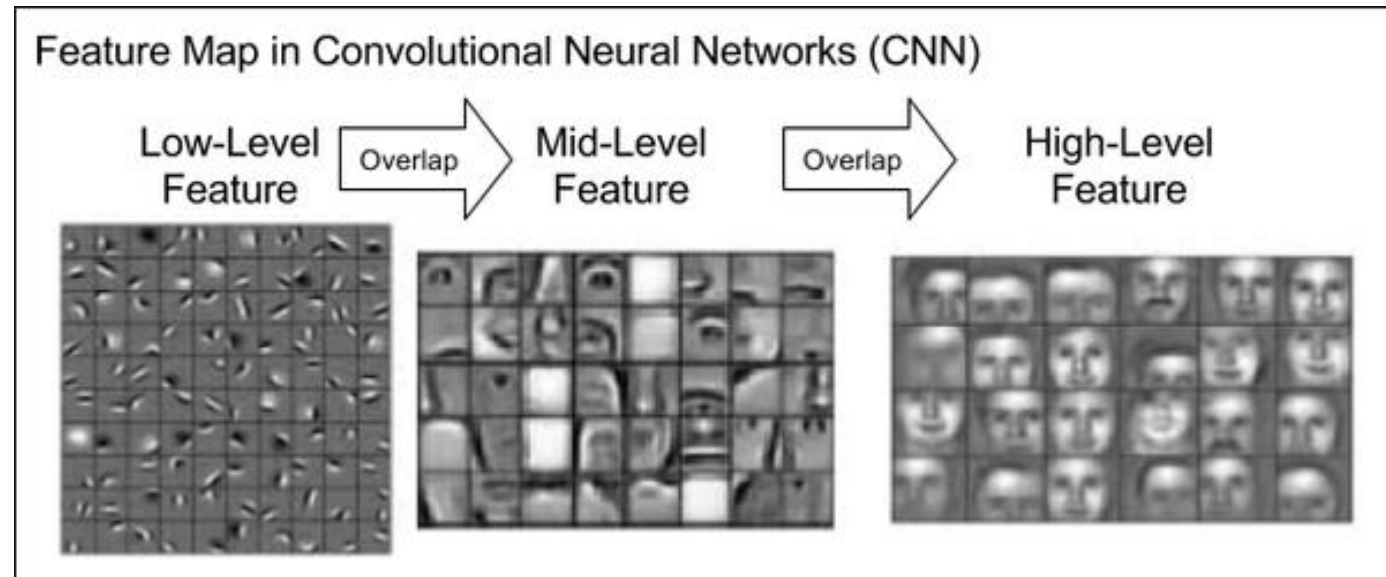
This would take their name and a series of photos of there face.



Processing the images

Using a Convolutional Neural Network (CNN)

CNN is most widely used for image processing because it is designed to process pixel data.



Tensorflow

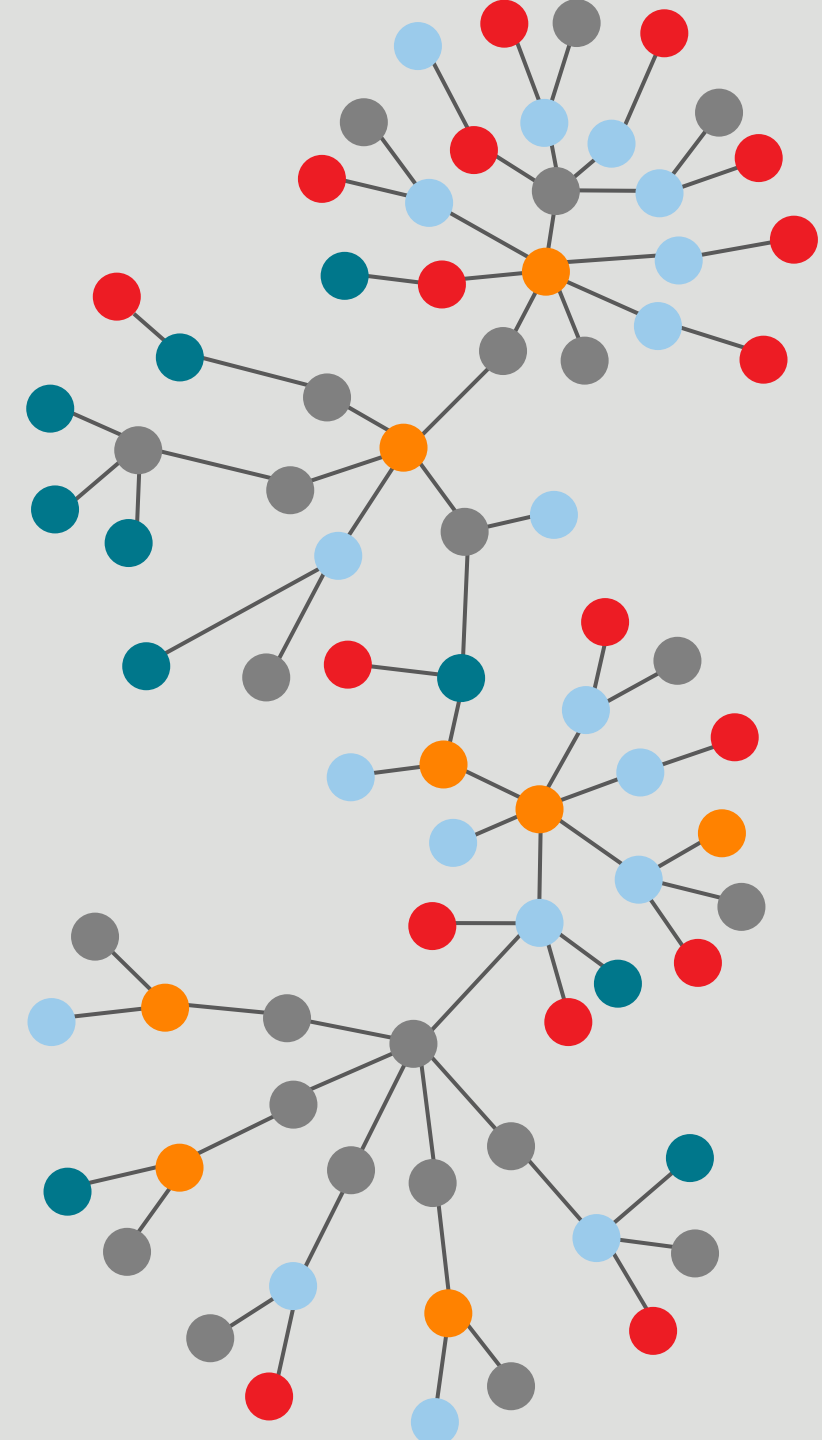
- Tensorflow is an open sourced artificial intelligence library that utilizes data flow graphs to build models.
- Tensorflow was the perfect solution for this project because it is meant to be used with large-scale neural networks.



Quick poll:

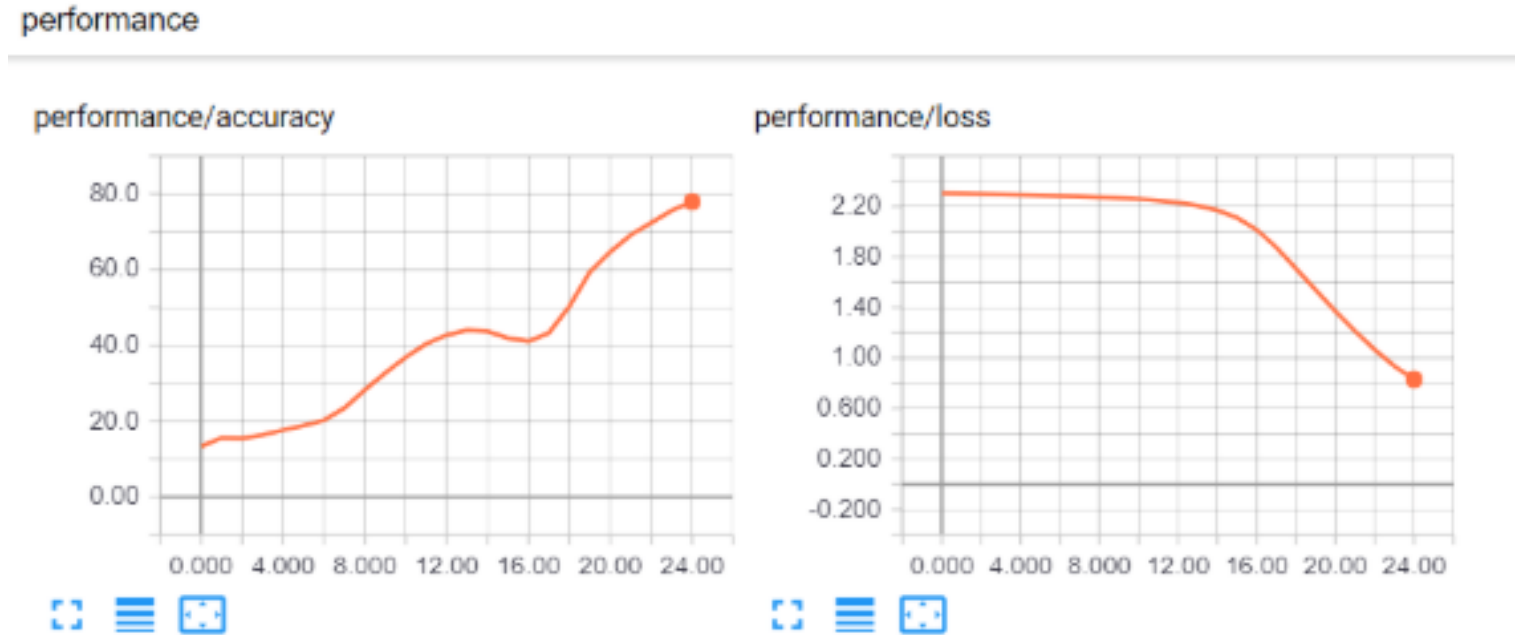
Are you familiar with the HPCC Systems GNN
BUNDLE?

See poll on bottom of presentation screen

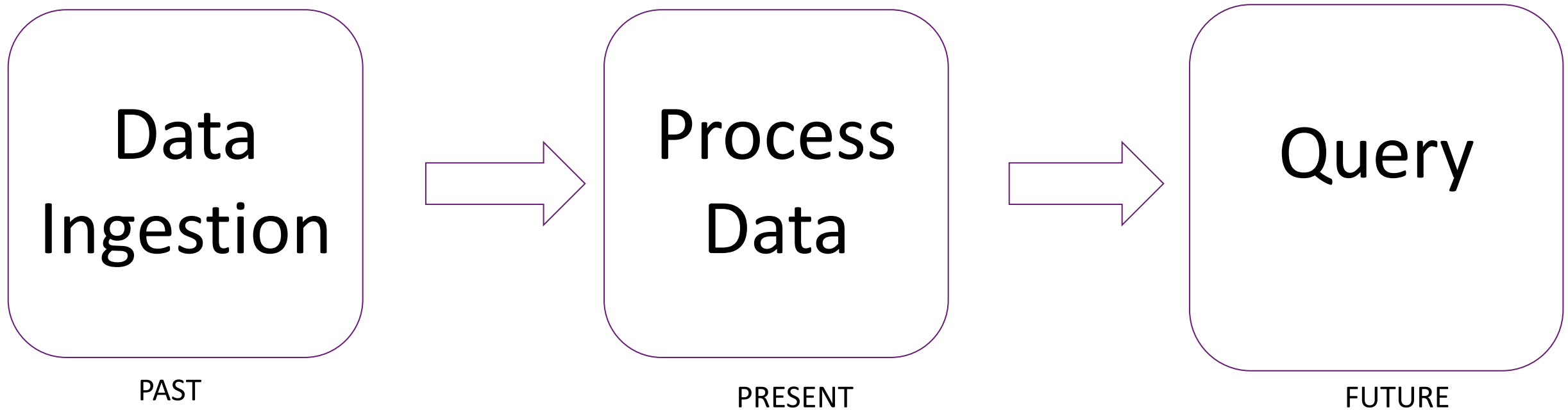


Learning To Use The GNN BUNDLE

- Testing Different Layering
- Testing different Datasets
- Ingesting Data From the Robot



Project Progression



Questions?

Jack Fields

High School Student

American Heritage School, Delray Beach, FL

bd551841@ahschool.com



More about the HPCC Systems Intern Program...

- Blog about the program: <https://hpccsystems.com/intern>
- Available projects: <https://hpccsystems.com/ideas-list>
- Student wiki: <https://hpccsystems.com/student-wiki>
- Read about our 2020 interns: <https://hpccsystems.com/blog/intro-interns2020>
- [Intern Contributions Past and Present](#) – Presentations, slides, posters and blogs
- 2020 program drawing to a close, 2021 program opens for proposals Oct/Nov – watch our [Student Programs Forum](#) for announcements

Submit a talk for an upcoming episode!

- Have a new success story to share?
- Want to pitch a new use case?
- Have a new HPCC Systems application you want to demo?
- Want to share some helpful ECL tips and sample code?
- Have a new suggestion for the roadmap?
- Be a featured speaker for an upcoming episode! Email your idea to Techtalks@hpccsystems.com
- Visit The Download Tech Talks wiki for more information: <https://hpccsystems.com/Tech-Talks>

Stay tuned for information on our next Tech Talk scheduled for **September 17, 2020** featuring more of our summer interns!

Thank You!



Visit our Tech Talk wiki for more information and to browse past episodes:
<https://hpccsystems.com/techtalks>