kitewheel
orchestrate great experiences

Production Deployment

Agenda

01 Build and Testing status

02 Metadata

03 Error Handling

04 Versioning

05 Pre-production checklist

06 Going Live

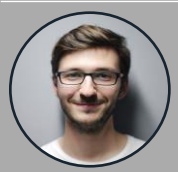
07 Escalation Support

08 Certification

Target Audiences

Primary

Configurer



Cory, the Configurer

INTERESTED IN…

- Identifying data sources
- Designing logic
- Solution architecting logic and rules
- Testing and deploying new orchestration journeys

Secondary

Technical & Support

Tanner, the Tech & Support

INTERESTED IN…

- Enable tech environment - servers, database
- Handle security and internet facing services
- Support accounts and projects
- Production support and continuous operation

Build and Testing Status

- Do you have at least a development and/or UAT environment?
- Have the graphs been tested fully in the development/UAT environment?
  - Unit Testing
  - Integration/ End-to-End Testing
  - User Acceptance Testing
  - Performance testing

Metadata

- Metadata in every graph
  For every transaction, all fields are recorded:
    - EnvironmentName: designated environment
    - GraphName: starting graph
    - ProjectName: name of project
    - TransactionId: unique ID associated to transaction
    - TransactionStart: starting time

- On error, the Error object in the Metadata is populated
    - Code: type of error (400, etc.)
    - Message: error message displayed in testing console
    - Node>Name: name of node that threw the error
    - Node>Type: type of node that threw the error
    - Time: time the error occurred (UTC)
    - TransactionState: state of schema and public variables at the time of the error

Error Handling Options

- Message Queues
  - Designated error handling message queues are written to on error events
  - Allow replay of the event/ trigger on the graph
  - This is useful if the services we are connecting to are not reliable

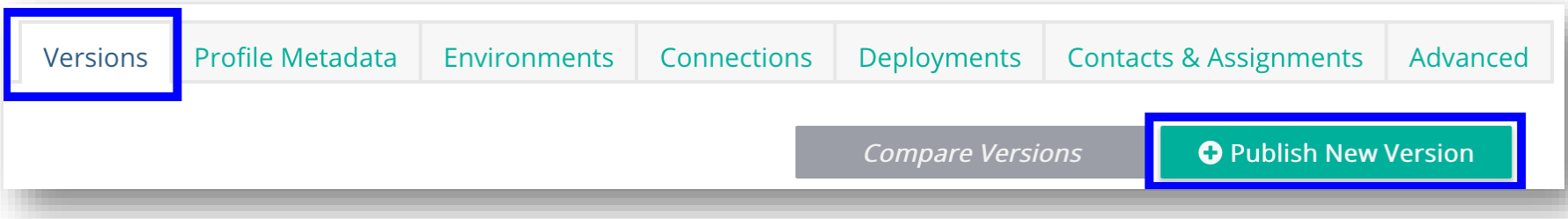
- Database
  - Log the error in an error log table with the error message, node and graph on error as well as the full schema
  - This information is found in the Metadata

- Calling an external Web Service
  - If needed, any external web service can be called to convey the error and details

Versioning

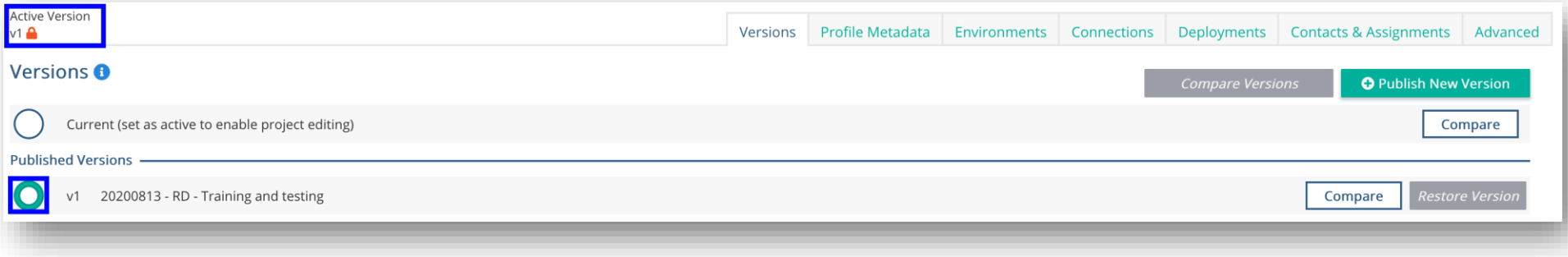
- Version control that allows snapshots to be created at any time of the **entire project**.
  - Note: this includes the journey map, graphs, project settings, etc.

- To create a new version:
  - Navigate to Project Settings screen
  - Choose "Versions" and click on the "Publish New Version" button
  - Enter the date YYYYMMDD, your initials and description of the version
  - Kitewheel will automatically append a version tag - 'v#' where # starts at 1



Choosing a Version

- In list of versions, choose a version by checking the box next to the version
- Once selected, the active version is shown in the editor
- The Current version is the current editable version

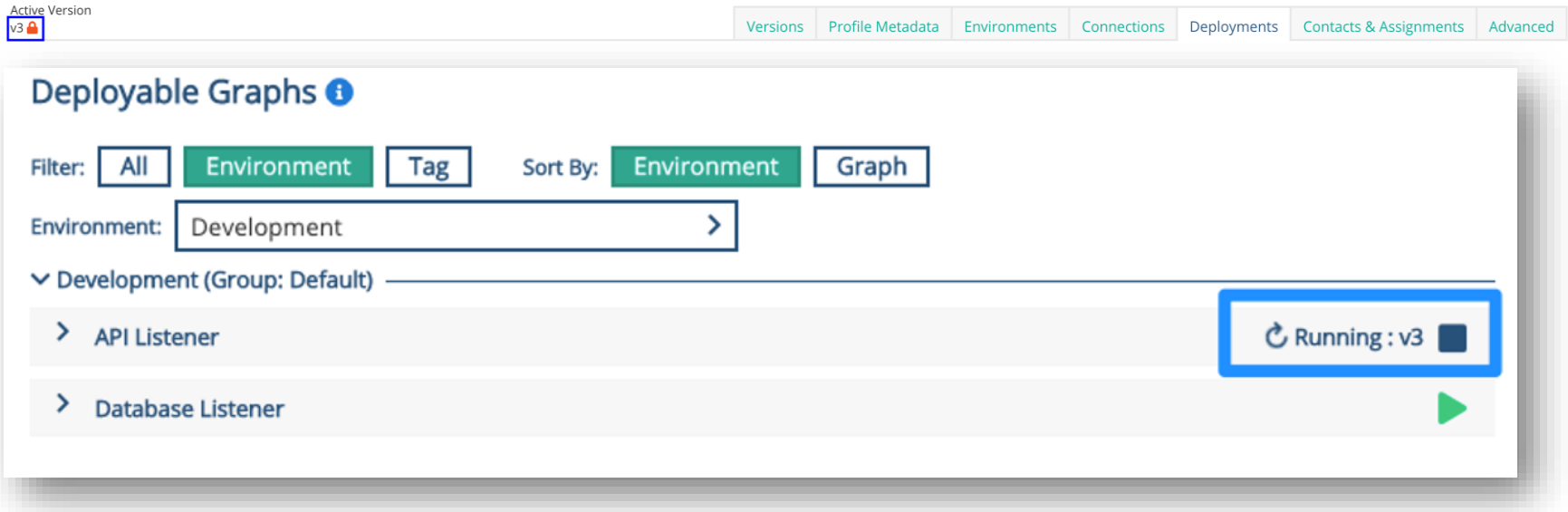


Navigating and Deploying a Version

- Navigating
  - When a version is selected, all data shown will be from this version (graphs, connections, environments, contacts)
  - When navigating in a version, the screen will be greyed out and the ability to edit will be disabled

- Deploying
  - When ready to deploy, if a published version is the active version when deploying a graph from admin screen, that version of the graph will be deployed, and will be indicated in UI
  - Two different versions cannot be deployed at the same time
    - All deployments must be stopped to deploy a new version

Navigating and Deploying a Version



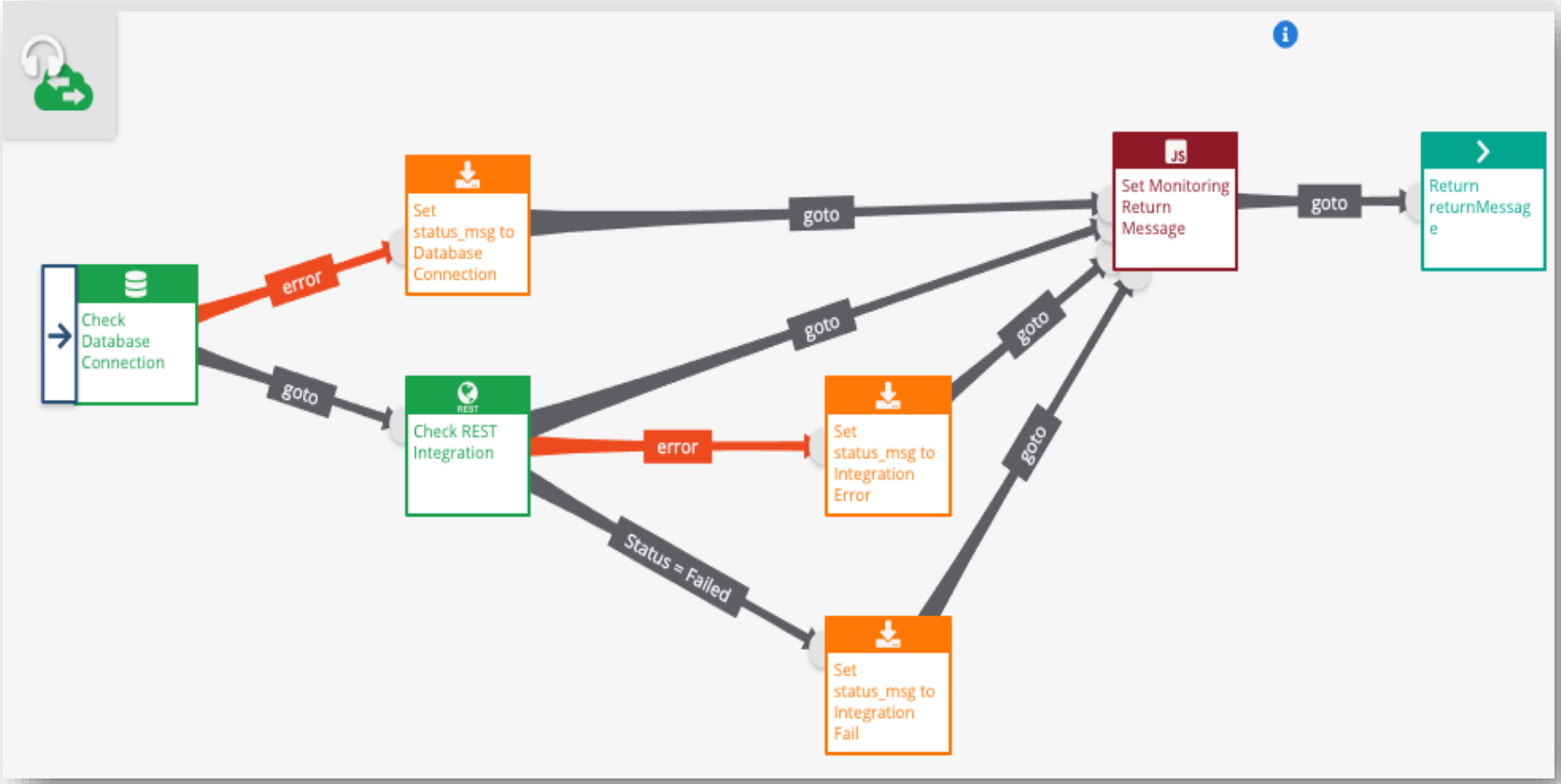
Monitoring the Project

- It is best practice to set up a monitoring endpoint in Kitewheel for all projects that are going live.
- The monitoring graph is a Kitewheel API graph which will run a simple test
  - Check REST endpoint(s)
  - Check database connection(s)
- Dependencies, important connections and any queues can be monitored.

Setting up Monitoring and Creating Health Routes

- Provide monitoring endpoint to Kitewheel Support
- Every X minutes, Kitewheel will check to ensure the dependent systems are available
- Provide instructions to Kitewheel Support when the endpoint throws an alert
- Please reach out to Kitewheel Support at least 2 weeks before the project goes live to set this up and have all details available at least 1 week before go live
  - Support Email: support@kitewheel.com
  - Web form: support.kitewheel.com

Monitoring Graph Example



Pre-production Checklist

- All production connections are configured and tested
- All necessary error handling is in place
- All testing is complete (Unit, UAT, E2E, etc.)
- Once testing is complete, a production ready version is created
- Traffic patterns in production are estimated to understand busy processing times throughout the week
- Monitoring is in place for the project
- Kitewheel Support is notified before going live

Going Live

- Once the checklist is complete, it is best practice to set up a go-live meeting (phone call, in-person, etc.) to ensure all parties are aligned
- Respective party will deploy the graphs
- Once live, conduct post-production tests (smoke testing, etc.) to ensure everything is working as intended

Escalation Support in Production

- The Hub logs at the engine level
- These logs are reviewed by the Kitewheel Support Team
- All Listener Graphs should also write out a transaction log to record application level processing information

Certification

- What will the Metadata section contain when an error occurs in a graph?
- What is a version and how is it created?
- What are the pre-production steps that have to be completed before deploying a graph?
- How many people are involved and informed before a production deployment?
- What are the various engine and listener settings that can be modified on a graph that has to be deployed?
- How are issues in Production detected and resolved?

kitewheel
orchestrate great experiences

Thank You

www.kitewheel.com