

git_comments:

1. !/bin/bash
2. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * <http://www.apache.org/licenses/LICENSE-2.0> * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
3. unmarshal the string to a JSON object
4. last line of stdout is the result JSON object as a string
5. program receives one argument: the JSON object as a string
6. can optionally log to stdout (or stderr)
7. * Licensed to the Apache Software Foundation (ASF) under one or more * contributor license agreements. See the NOTICE file distributed with * this work for additional information regarding copyright ownership. * The ASF licenses this file to You under the Apache License, Version 2.0 * (the "License"); you may not use this file except in compliance with * the License. You may obtain a copy of the License at * * <http://www.apache.org/licenses/LICENSE-2.0> * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License.
8. +build integration
9. wskaction.Annotations = listOfOutputs
10. Create action data object from client library
11. when an action has Docker image specified, set exec.Kind to "blackbox" and set exec.Image to specified image e.g. dockerhub/image when an action Native is set to true, set exec.Image to openwhisk/skeleton
12. Web Export
13. update the action (of type Action) to set its name here key name is the action name
14. Action.Inputs
15. **comment:** TODO() use LIMITS_UNSUPPORTED in yamlparser to enumerate through instead of hardcoding emit warning errors if these limits are not nil
label: code-design
16. **comment:** Action.Outputs TODO{} add outputs as annotations (work to discuss officially supporting for compositions)
label: code-design
17. Check if action.Function is specified with action.Code with action.Code, action.Function is not allowed with action.Code, action.Runtime should be specified
18. Action.Limits
19. * Action.Runtime * Perform few checks if action runtime is specified in manifest YAML file * (1) Check if specified runtime is one of the supported runtimes by OpenWhisk server * (2) Check if specified runtime is consistent with action source file extensions * Set the action runtime to match with the source file extension, if wskdeploy is not invoked in strict mode
20. determine extension of the given action source file
21. Action.Annotations
22. validate manifest_parser.ComposeActions() method

git_commits:

1. **summary:** Adding Docker and Native support (#801)
message: Adding Docker and Native support (#801) * refactoring composeActions - phase 1 * refactor compose action - phase 2 * refactor composeAction - phase 3 * refactor composeaction - phase 4 * refactor composeAction - phase 5 * adding docker and native support * adding license header * refactor manifest parser to utilize common inputs and annotations function * fixing compile error * using PATH SEPARATOR as rune * capitalizing Packages in dependency - fixing unit test * updating docker unit test - error messages * moving action file name to a var
label: code-design

github_issues:

1. **title:** Support "blackbox" Docker images
body: This has been a discussion since last Sept. and asked Daisy to look into supporting at that time. Need to track this work via this issue. `` Sept 2016: Daisy: Blackbox (Docker) Actions: Need to think about how to set the field "runtime" if it is a docker action. We should document it. ``
2. **title:** Support "blackbox" Docker images
body: This has been a discussion since last Sept. and asked Daisy to look into supporting at that time. Need to track this work via this issue. `` Sept 2016: Daisy: Blackbox (Docker) Actions: Need to think about how to set the field "runtime" if it is a docker action. We should document it. ``
label: code-design
3. **title:** Support "blackbox" Docker images
body: This has been a discussion since last Sept. and asked Daisy to look into supporting at that time. Need to track this work via this issue. `` Sept 2016: Daisy: Blackbox (Docker) Actions: Need to think about how to set the field "runtime" if it is a docker action. We should document it. ``
4. **title:** Support "blackbox" Docker images
body: This has been a discussion since last Sept. and asked Daisy to look into supporting at that time. Need to track this work via this issue. `` Sept 2016: Daisy: Blackbox (Docker) Actions: Need to think about how to set the field "runtime" if it is a docker action. We should document it. ``
5. **title:** Support "blackbox" Docker images
body: This has been a discussion since last Sept. and asked Daisy to look into supporting at that time. Need to track this work via this issue. `` Sept 2016: Daisy: Blackbox (Docker) Actions: Need to think about how to set the field "runtime" if it is a docker action. We should document it. ``

github_issues_comments:

1. See: <https://github.com/apache/incubator-openwhisk-wskdeploy/issues/279>
2. **body:** TODO: look at CLI conventions for what they call this? I believe its "docker" so we would then use: runtime: docker then we would have to be able to validate the image reference exists/is valid (DockerHub) and if we need to add it to an existing key or new key in the YAML manifest. Need a recommendation here on what is clear/friendly to the user. Then we have to consider "private" Docker repos (hubs) and credentials. Please.... also make sure this would work in the future for Rocket images.
label: code-design
3. Private docker repos are not yet supported by openwhisk, see <https://github.com/apache/incubator-openwhisk/issues/1362>

github_pulls:

1. **title:** Adding Docker and Native support
body: Closes #623 Closes #279 List of changes in this PR: 1. Major Refactoring of `manifest_parser.go`:
* Replace `str != ""` kind of check with `len(str) != 0` * Replace "/" with constant `PATH_SEPARATOR`
* Created a common function to parse inputs and outputs, updated all `Compose*` routines to utilize this common function. Same with annotations. * Split `ComposeAction` function into smaller routines with validating code, validating function, reading code, reading function, creating action exec depending on action function and code, etc. 2. Added support for docker and native actions with unit test and integration test.
2. **title:** Adding Docker and Native support
body: Closes #623 Closes #279 List of changes in this PR: 1. Major Refactoring of `manifest_parser.go`:
* Replace `str != ""` kind of check with `len(str) != 0` * Replace "/" with constant `PATH_SEPARATOR`
* Created a common function to parse inputs and outputs, updated all `Compose*` routines to utilize this common function. Same with annotations. * Split `ComposeAction` function into smaller routines with validating code, validating function, reading code, reading function, creating action exec depending on action function and code, etc. 2. Added support for docker and native actions with unit test and integration test.
3. **title:** Adding Docker and Native support
body: Closes #623 Closes #279 List of changes in this PR: 1. Major Refactoring of `manifest_parser.go`:
* Replace `str != ""` kind of check with `len(str) != 0` * Replace "/" with constant `PATH_SEPARATOR`
* Created a common function to parse inputs and outputs, updated all `Compose*` routines to utilize this common function. Same with annotations. * Split `ComposeAction` function into smaller routines with

validating code, validating function, reading code, reading function, creating action exec depending on action function and code, etc. 2. Added support for docker and native actions with unit test and integration test.

4. **title:** Adding Docker and Native support

body: Closes #623 Closes #279 List of changes in this PR: 1. Major Refactoring of `manifest_parser.go`:
* Replace `str != ""` kind of check with `len(str) != 0` * Replace "/" with constant `PATH_SEPARATOR`
* Created a common function to parse inputs and outputs, updated all `Compose*` routines to utilize this common function. Same with annotations. * Split `ComposeAction` function into smaller routines with validating code, validating function, reading code, reading function, creating action exec depending on action function and code, etc. 2. Added support for docker and native actions with unit test and integration test.

5. **title:** Adding Docker and Native support

body: Closes #623 Closes #279 List of changes in this PR: 1. Major Refactoring of `manifest_parser.go`:
* Replace `str != ""` kind of check with `len(str) != 0` * Replace "/" with constant `PATH_SEPARATOR`
* Created a common function to parse inputs and outputs, updated all `Compose*` routines to utilize this common function. Same with annotations. * Split `ComposeAction` function into smaller routines with validating code, validating function, reading code, reading function, creating action exec depending on action function and code, etc. 2. Added support for docker and native actions with unit test and integration test.

6. **title:** Adding Docker and Native support

body: Closes #623 Closes #279 List of changes in this PR: 1. Major Refactoring of `manifest_parser.go`:
* Replace `str != ""` kind of check with `len(str) != 0` * Replace "/" with constant `PATH_SEPARATOR`
* Created a common function to parse inputs and outputs, updated all `Compose*` routines to utilize this common function. Same with annotations. * Split `ComposeAction` function into smaller routines with validating code, validating function, reading code, reading function, creating action exec depending on action function and code, etc. 2. Added support for docker and native actions with unit test and integration test.

7. **title:** Adding Docker and Native support

body: Closes #623 Closes #279 List of changes in this PR: 1. Major Refactoring of `manifest_parser.go`:
* Replace `str != ""` kind of check with `len(str) != 0` * Replace "/" with constant `PATH_SEPARATOR`
* Created a common function to parse inputs and outputs, updated all `Compose*` routines to utilize this common function. Same with annotations. * Split `ComposeAction` function into smaller routines with validating code, validating function, reading code, reading function, creating action exec depending on action function and code, etc. 2. Added support for docker and native actions with unit test and integration test.

label: code-design

8. **title:** Adding Docker and Native support

body: Closes #623 Closes #279 List of changes in this PR: 1. Major Refactoring of `manifest_parser.go`:
* Replace `str != ""` kind of check with `len(str) != 0` * Replace "/" with constant `PATH_SEPARATOR`
* Created a common function to parse inputs and outputs, updated all `Compose*` routines to utilize this common function. Same with annotations. * Split `ComposeAction` function into smaller routines with validating code, validating function, reading code, reading function, creating action exec depending on action function and code, etc. 2. Added support for docker and native actions with unit test and integration test.

label: documentation

9. **title:** Adding Docker and Native support

body: Closes #623 Closes #279 List of changes in this PR: 1. Major Refactoring of `manifest_parser.go`:
* Replace `str != ""` kind of check with `len(str) != 0` * Replace "/" with constant `PATH_SEPARATOR`
* Created a common function to parse inputs and outputs, updated all `Compose*` routines to utilize this common function. Same with annotations. * Split `ComposeAction` function into smaller routines with validating code, validating function, reading code, reading function, creating action exec depending on action function and code, etc. 2. Added support for docker and native actions with unit test and integration test.

10. **title:** Adding Docker and Native support

body: Closes #623 Closes #279 List of changes in this PR: 1. Major Refactoring of `manifest_parser.go`:
* Replace `str != ""` kind of check with `len(str) != 0` * Replace "/" with constant `PATH_SEPARATOR`
* Created a common function to parse inputs and outputs, updated all `Compose*` routines to utilize this common function. Same with annotations. * Split `ComposeAction` function into smaller routines with validating code, validating function, reading code, reading function, creating action exec depending on

action function and code, etc. 2. Added support for docker and native actions with unit test and integration test.

11. **title:** Adding Docker and Native support

body: Closes #623 Closes #279 List of changes in this PR: 1. Major Refactoring of `manifest_parser.go`:
* Replace `str != ""` kind of check with `len(str) != 0` * Replace "/" with constant `PATH_SEPARATOR`
* Created a common function to parse inputs and outputs, updated all `Compose*` routines to utilize this common function. Same with annotations. * Split `ComposeAction` function into smaller routines with validating code, validating function, reading code, reading function, creating action exec depending on action function and code, etc. 2. Added support for docker and native actions with unit test and integration test.

12. **title:** Adding Docker and Native support

body: Closes #623 Closes #279 List of changes in this PR: 1. Major Refactoring of `manifest_parser.go`:
* Replace `str != ""` kind of check with `len(str) != 0` * Replace "/" with constant `PATH_SEPARATOR`
* Created a common function to parse inputs and outputs, updated all `Compose*` routines to utilize this common function. Same with annotations. * Split `ComposeAction` function into smaller routines with validating code, validating function, reading code, reading function, creating action exec depending on action function and code, etc. 2. Added support for docker and native actions with unit test and integration test.

13. **title:** Adding Docker and Native support

body: Closes #623 Closes #279 List of changes in this PR: 1. Major Refactoring of `manifest_parser.go`:
* Replace `str != ""` kind of check with `len(str) != 0` * Replace "/" with constant `PATH_SEPARATOR`
* Created a common function to parse inputs and outputs, updated all `Compose*` routines to utilize this common function. Same with annotations. * Split `ComposeAction` function into smaller routines with validating code, validating function, reading code, reading function, creating action exec depending on action function and code, etc. 2. Added support for docker and native actions with unit test and integration test.

14. **title:** Adding Docker and Native support

body: Closes #623 Closes #279 List of changes in this PR: 1. Major Refactoring of `manifest_parser.go`:
* Replace `str != ""` kind of check with `len(str) != 0` * Replace "/" with constant `PATH_SEPARATOR`
* Created a common function to parse inputs and outputs, updated all `Compose*` routines to utilize this common function. Same with annotations. * Split `ComposeAction` function into smaller routines with validating code, validating function, reading code, reading function, creating action exec depending on action function and code, etc. 2. Added support for docker and native actions with unit test and integration test.

github_pulls_comments:

1. @ddragosd @pritidesai In order to support private (with credentials), we would need to add support for the credential and repository schema (envisioned in spec.). Priti, do you agree with this assessment? If so, can we open an epic/issues for this and prioritize?
2. This is not supported in core, the invoker doesn't deal with credentials to pull the image it just executes docker pull Cc @rabbah
3. @mrutkows I think its beyond the scope of `wskdeploy`. We have repository and credentials defined [here](https://github.com/apache/incubator-openwhisk-wskdeploy/blob/master/specification/openwhisk_v0.9.2.pdf) on line 558 and line 571. But with docker actions the way they are designed, images are not pulled during action creation. So even if we have access to credentials of private repo and we could pass that to OpenWhisk server but its not supported by server side. We can address this by what Rodric has commented here https://github.com/apache/incubator-openwhisk/issues/1362 saying "One way to support this is by pulling images from the private registry to a whisk managed registry at action create time. This would avoid some the challenges around key management for docker login. In general I think this is a good approach for docker actions as it will make it easier for vetting images; but not without its demerits." We can definitely add support for this in `wskdeploy`, is that what we are looking for? @mrutkows @ddragosd @csantanapr @rabbah suggestions please?
4. Yep this is something that wskdeploy can't solve at the moment

github_pulls_reviews:

1. Nice unit test we have needed for some time...

2. **body:** Would have preferred this declared at top of function so we can quickly identify test data files used within the test method and they do not scroll off the right edge.
label: code-design
3. **body:** we will get you yet typo! ...
label: documentation
4. Do we need the source code for the "exec" binary? Is this example.c?
5. great 👍 should we assume that custom/private docker registries are supported as well, provided the machine has the credentials needed to connect to that registry ?
6. This is a sample in Go from https://console.bluemix.net/docs/openwhisk/openwhisk_actions.html#creating-go-actions. I have three examples included, (1) a dir with exec in bash (`function: <dir>`) (2) a dir with exec in go (compiled) (`function: <dir>`) (3) exec.zip. Its pretty much same workflow for example.c
7. @ddragosd yes and no, custom docker registries are supported but not private. OpenWhisk does not support private registries yet (see <https://github.com/apache/incubator-openwhisk/issues/1362>) since the docker image is pulled when you invoke that action for the first time so even if the machine from where the action was created has credentials, its not useful.
8. 👍 done

jira_issues:

jira_issues_comments: