**Title**:

# **Installation of Private Docker-Registry application**

**Description**:

Workflow for handling packages related to the installation of Private Docker-Registry application. This bundle has dependency on Core and Validation KIs bundles. For more information related to Docker-Registry and installation process, please check<https://www.digitalocean.com/community/tutorials/how-to-set-up-a-private-docker-registry-on-ubuntu-14-04>

**Readme**

**Important**: the Knowledge Items of this bundle try to simulate Workflows seen in frameworks such as Chef, Puppet and Ansible.

For this reason, Workflow KIs seem to not follow traditional best practices of KI Creation in terms of reusability. They are, however atomic pieces of knowledge, meaning the knowledge cannot be broken down further while still maintaining the same level of functionality.

Due to the controlled nature of the Workflow approach, the Workflow KIs try to structure Tasks like steps in a sequence rather than letting AutoPilot pick the path to a solution.

In order to simulate these Workflows, the Knowledge Items are split into 3 groups:

* Workflow KIs
* Validation KIs and
* Core KIs.

This bundle intends to install Docker Registry with all its related application and libraries on the given host.

**Dependencies**

1. Core KIs bundle
2. Validation KIs bundle

For this bundle to run, you will need MARS nodes in AutoPilot engine. For your reference, please see sample nodes below (at the end of this readme).

**Steps to run**

1. Put the validation KIs bundle in AutoPilot engine
2. Put the core KIs bundle in AutoPilot engine
3. Put the Workflow KI DockerRegistryWorkflowHandlePKG.xml (attached with this bundle) in AutoPilot engine
4. Put the Issue in AutoPilot engine to trigger this KI
   * For your reference, please see sample issue below
5. This will trigger the KI and run the workflow to install Docker Registry on your machine (as specified in the Issue)

Workflow KI in this bundle, runs below installation steps (as per the link mentioned in the description) on the target machine (as specified in the Issue)

1. Install pre-requisites package build-essential, python-dev , libevent-dev,

python-pip, liblzma-dev , swig .

1. Install docker registry through pip.
2. Create config.yml file and a directory in var named docker-registry
3. Replace the /tmp to /var in config.yml
4. Create a directory named docker-registry
5. Write a file to execute the command
6. Start the docker-registry service

**Sample Issue**

Issue:

NodeID: OpexSoftware:Workflow:Application:EnterpriseInfrastructure

xmlns: http://www.arago.de/IssueSchema

IssueSubject: 'Installing Docker Registry'

DockerRegistryWorkflowHandlePKG:

TargetApp: EnterpriseInfrastructure

TargetMachine: BD

TargetState: Created

User: bhagyashree

Host: 192.168.1.89

**Machine Node**

<Linux xmlns="http://mars-o-matic.com" ID="OpexSoftware:Workflow:Machine:BD"

HasAgentType\_WatchMe="False" MachineArchitecture="x86\_64"

NodeType="Machine" MachineClass="Linux" NodeName="BD">

<Dependencies>

<Node ID="OpexSoftware:Workflow:Software:Docker" />

</Dependencies>

<CustomerInformation ID="opex.com" Name="opex"/>

<Extensions>

<ssh User="bhagyashree" />

<PasswordPolicy MinLength="8" MaxLength="14" CharSet="alnum" />

</Extensions>

<OSInformation Name="Ubuntu" MajorVersion="14" Vendor="Ubuntu" />

<NetworkInformation>

<InterfaceInformation>

<Interface Name="eth1" IP="192.168.1.89" />

</InterfaceInformation>

</NetworkInformation>

</Linux>

**Software Node**

<AutoPilot xmlns="http://mars-o-matic.com" ID="OpexSoftware:Workflow:Software:Docker"

NodeType="Software" SoftwareClass="Automation"

SoftwareSubClass="AutoPilot" NodeName="Docker">

<Dependencies>

<Node ID="OpexSoftware:Workflow:Machine:BD" />

<Node ID="OpexSoftware:Workflow:Resource:WebResource" />

</Dependencies>

<CustomerInformation ID="opex.com" Name="opex"/>

</AutoPilot>

**Resource Node**

<Service xmlns="http://mars-o-matic.com" ID="OpexSoftware:Workflow:Resource:WebResource"

NodeType="Resource" ResourceClass="Service" NodeName="WebResource">

<Dependencies>

<Node ID="OpexSoftware:Workflow:Software:Docker" />

<Node ID="OpexSoftware:Workflow:Application:EnterpriseInfrastructure" />

</Dependencies>

<CustomerInformation ID="opex.com" Name="OpexSoftware" />

</Service>

**Application Node**

<EnterpriseInfrastructure xmlns="http://mars-o-matic.com" ID="OpexSoftware:Workflow:Application:EnterpriseInfrastructure"

NodeType="Application" ApplicationClass="Enterprise"

ApplicationSubClass="EnterpriseInfrastructure" NodeName="EnterpriseInfrastructure">

<Dependencies>

<Node ID="OpexSoftware:Workflow:Resource:WebResource" />

</Dependencies>

<CustomerInformation ID="opex.com" Name="OpexSoftware" />

</EnterpriseInfrastructure>