

CANONICAL



Juju Administrator

Training Curriculum

Prepared For:	Canonical
Prepared By:	Jon Seager
Prepared On:	22 June 2021
Version:	1.0

Introduction & Scope

Beginning with an introduction to the purpose and history of Juju, model-driven operations and how application management can transform the way businesses manage applications, the Juju Administrator course is designed to enable administrators to quickly become proficient at managing Juju and deploying, configuring and integrating applications across computing substrates. The course is a blend of theory sessions and hands-on practical labs which see attendees deploying charmed applications on VMs, containers and Kubernetes clusters.

The duration of the course is three days.

Agenda

Day 1

Juju Basics

Agenda

Section	Notes	Time
Introduction	<p>Basic introduction to Juju, terminology such as "Charmed Operator", and "Operator Lifecycle Manager"</p> <ul style="list-style-type: none">- What is Juju?- What problems does Juju solve?- Comparison to other tools/methodologies<ul style="list-style-type: none">- Ansible/Chef/Puppet- Terraform/Pulumi- Generic operator pattern- Operator Framework- Juju's consistency across substrates<ul style="list-style-type: none">- Bare-metal/MAAS/LXD- Azure/AWS/GCP/Oracle/Rackspace- VMWare- Openstack- Kubernetes- Deployment patterns and examples<ul style="list-style-type: none">- Applications/stacks on single substrates- Applications and services across multiple substrates- Charmed <Openstack Kubernetes OSM>- Juju Community<ul style="list-style-type: none">- Docs- Discourse/Mattermost- Charmhub- Github/Launchpad Repos- Rest of course outline- Take questions on any of the above	1 hour
Demonstration	Deploy the Charmed Kubernetes Lite bundle using the Juju dashboard. This is a suitably complex example that brings into focus the simplicity that Juju can bring to complex deployments and illustrates the visual aspect of application modelling	10 mins
Concepts	<p>This section should crystallize the definitions of the various terms used when describing Juju and Charmed Operators.</p> <ul style="list-style-type: none">- Controllers<ul style="list-style-type: none">- Controller function- Juju CLI interaction with controller API- Controller database and state storage- Single-substrate/multi-substrate controllers	1 hr 30 mins

	<ul style="list-style-type: none"> - JAAS / JIMM - Charms - Models - Applications - Units - Leadership - Charmed Operators - Events - Actions - Config - Bundles - Take questions on any of the above 	
Break / Informal Q&A	Juju is a lot, give people a break ;)	10 mins
Advanced Concepts #1	<p>This section should illustrate the more advanced features of Juju and start to bring into focus the complex problems it can solve:</p> <ul style="list-style-type: none"> - Deployment constraints (memory, cpu, etc.) - Juju users and credentials - High-availability controllers - Subordinate charms - Relations <ul style="list-style-type: none"> - Basics on relations and application integration - Interfaces - Relations across different apps (provides/requires) - Peer relations - Relation data and persistence - Cross Model Relations - Take questions on any of the above 	1 hr
Demonstration	<ul style="list-style-type: none"> - Create a new Juju controller - Deploy wordpress (requires database) - Deploy mysql (provides database) - Create the relation between the two applications, and practically explain the interaction via relations (provides/requires). 	20 mins
Break / Informal Q&A	Break	10 mins
Advanced Concepts #2	<ul style="list-style-type: none"> - Libraries - Juju on Kubernetes <ul style="list-style-type: none"> - Highlight similarities, not differences - Introduce Pebble 	30 mins
Demonstration	Deploy Charmed Kubeflow Lite on MicroK8s and explore the deployment both in the Juju CLI, Juju Dashboard and using `kubectl`	30 mins

Juju Administrator Part 1

Agenda

Section	Notes	Time
Introduction	<p>This section should be used to introduce the course material, explain the outcomes and what will be achieved in the labs</p> <ul style="list-style-type: none">- Verify prior knowledge/Juju Basics attendance- Verify that students have a suitable lab setup<ul style="list-style-type: none">- Introduce Multipass for creating VMs quickly on physical hardware if relevant- Talk through installing Multipass, LXD, Juju, MicroK8s as required- Outline the rest of the course	15 mins
Adding clouds and bootstrapping	<ul style="list-style-type: none">- Brief recap about Juju Controllers and their purpose- Explain the purpose of `add-cloud/add-credential`- Explain the purpose of `bootstrap`- Explain what <i>actually</i> happens during a bootstrap on LXD- Explain what <i>actually</i> happens during a bootstrap on <customer preferred cloud(s)>	20 mins
Lab 1	<ul style="list-style-type: none">- Provision a new VM with Multipass- Install and init LXD- Bootstrap a Juju controller called `lxd01` on LXD- Bootstrap a Juju controller called `micro` on MicroK8s- Take questions on any of the above until most students finish the lab task	30 mins
Juju Client Config	<ul style="list-style-type: none">- Cover how the juju client is configured (series of files in `~/.local/share/juju`)- How to backup the Juju client- What the purpose of the config files is	15 mins
Q&A / Wrap up	<p>Recap of key messages:</p> <ul style="list-style-type: none">- What problems does Juju solve- Model-driven ops- Where to find docs- Where to learn more/community	25 mins

Day 2

Juju Administrator Part 2

Agenda

Section	Notes	Time
Applications and Models	<ul style="list-style-type: none">- Briefly recap applications, units, leadership- Introduce Charmhub and application versions/channels/revisions<ul style="list-style-type: none">- Show on charmhub- Illustrate the `juju info <charm>` command- Introduce the `juju models` command- Demonstrate an application deployment and removal- Discuss application states (active, waiting, blocked, maintenance)- Introduce the `juju status` command<ul style="list-style-type: none">- Show the detail given when output as YAML/JSON (charm revision, uuid, etc)- Download a charm using `juju download` and open in an archive manager, explain the contents of the charm (metadata.yaml, etc)- Explain the how applications are formed on Kubernetes and the sidecar pattern	1 hr
Lab 2	<ul style="list-style-type: none">- Create a new model on the LXD controller called `hello-juju`- Deploy the `hello-juju` application on LXD- Explore with `juju status`- Explore with `lxc list`- Validate access to the webpage- Scale application up to 3 units, and back down to 1 unit- Switch models to a MicroK8s model- Deploy the `hello-kubecon` Kubernetes application<ul style="list-style-type: none">- Explore the deployment with kubectl- Inspect the running containers in the pod- Take questions on any of the above until most students finish the lab task	30 mins
Application Scaling and Constraints	<ul style="list-style-type: none">- Illustrate deploying applications to specific machines/containers with `juju deploy --to`- Introduce the `show-status-log` command- Introduce application constraints<ul style="list-style-type: none">- Defined at bootstrap, deploy or add-machine- Defined at model or controller scope- Highlight differences between constraints on public cloud, LXD and Kubernetes (max vs min)- Illustrate setting default model constraints- Deploy a machine with specific memory and cpu constraints- Show how to query constraints ('get-constraints') for an application and set post-deploy with 'set-constraints'.	30 mins
Lab 3	<ul style="list-style-type: none">- Deploy `hello-juju` to a fresh model on LXD- Scale the `hello-juju` application to 3 units	30 mins

	<ul style="list-style-type: none"> - Set the default model constraints to `mem=2G` and `cpu=2` - Remove `hello-juju` and redeploy, specifying different constraints - Get the constraints from the recently deployed `hello-juju` app - Update the constraints then scale the application - Switch to the MicroK8s model and scale the `hello-kubecon` app, pay attention to how the operator has scaled with the application - Take questions on any of the above until most students finish the lab task 	
Break	Break / Informal Q&A	10 mins
Application Config and Actions	<ul style="list-style-type: none"> - Briefly recap application config, where it is stored and how it is set - Show how config items can be set using the CLI <ul style="list-style-type: none"> - At deploy time - When deployed - Illustrate dumping config from running charm as JSON/YAML etc at the command line - Show how config can be passed as a YAML file - Illustrate config documentation for a charm on charmhub - Recap charm actions and their purpose - Demonstrate `juju actions`, `juju run-action`, `juju show-action-output` etc 	30 mins
Lab 4	<ul style="list-style-type: none"> - Configure `hello-juju` with a new port and site by specifying options at the command line - Reconfigure those options by defining the options in a YAML file and passing it in - Discover the actions associated with the `hello-kubecon` application. Run the action and inspect the results. - Take questions on any of the above until most students finish the lab task 	30 mins
Machine Access	<ul style="list-style-type: none"> - Explain how machines can be accessed 'in-band' with `juju ssh` and `juju scp` - Demonstrate accessing a machine by unit name - Demonstrate copying a file into a machine - Show executing one-time commands and getting a shell - Use `juju run` to execute a command - Talk about accessing out of band <ul style="list-style-type: none"> - SSH directly - `lxc exec` - With virsh/virt-manager if the machines are VMs not LXC containers - Briefly demonstrate the `juju add-machine` command - add a new machine and another existing machine over SSH (use multipass) - Switch to the MicroK8s model and use both `juju ssh` and `kubectl exec` to gain shell access to the charm container in the `hello-kubecon-0` pod 	30 mins
Lab 5	<ul style="list-style-type: none"> - Access the hello-juju machine using `juju ssh` - Find the name of the container using `juju status` and access directly using `lxc exec` - Add your public key to the machine using a shell gained using `juju ssh` then ssh in using just `ssh user@ip` 	15 mins
Break	Break / Informal Q&A	10 mins

Logging	<ul style="list-style-type: none"> - Sources of log information - Location of logs on disk - Use of the `juju debug-log` command <ul style="list-style-type: none"> - Log levels - Filter by application/unit - Location of agent logs on machines - Adjusting the logging level for controllers and applications with `juju model-config` 	20 mins
Lab 6	<ul style="list-style-type: none"> - Set the model-config to include debug logs for units - Remove the `hello-juju` application - Redeploy the `hello-juju` application - Make a configuration change - Observe the logs in `juju debug-log` - Filter the `juju debug-log` results to only show controller logs - Filter the `juju debug-log` results to only show hello-juju logs - Use `juju scp` to copy the agent log file from `hello-juju/0` to your local machine - Take questions on any of the above until most students finish the lab task 	30 mins
Juju Hook Tools & Debugging	<ul style="list-style-type: none"> - What are Juju hook tools? - List the hook tools and explain them - Explain they are actually all contained in `jujuc` and symlinked, similar to how busybox works - Explain how they are invoked in the operator framework - How to run a hook tool using `juju debug-hooks` - How to run a hook tool using `juju run` - When you might need to run a hook tool 	30 mins
Lab 7	<ul style="list-style-type: none"> - Use `juju run` and the `unit-get` hook tool to get the private IP address of the deployed hello-juju application - Get the status of the `hello-juju/0` unit using the `status-get` hook tool - Set the status of the hello-juju application with the `status-set` hook tool - Get the config of the hello-juju application using the `config-get` hook tool in JSON format - Write a debug log message into the juju log using a hook tool - Set an item in state, then update it, then delete it. Check its value along the way - Take questions on any of the above until most students finish the lab task 	30 mins

Day 3

Juju Administrator Part 3

Agenda

Section	Notes	Time
Relations	<ul style="list-style-type: none">- Cover relation concepts again<ul style="list-style-type: none">- Relation interfaces- Relation data flow- Hooks fired in the charm- Relation data persistence- Illustrate how relation data is built up- Talk about peer relations<ul style="list-style-type: none">- Explain exchanging data between peers using relation data in the context of a clustered application eg. mongo or elastic- Introduce subordinate charms and how they are related to primaries<ul style="list-style-type: none">- Illustrate deployment of a subordinate charm- Show how the subordinate scales with the primary- Demonstrate how to establish relations explicitly (defining the interfaces) and implicitly where there is no ambiguity- Talk about cross-model relations and explain the use cases	1 hr
Lab 8	<ul style="list-style-type: none">- Deploy postgres and mattermost on microk8s and relate the two applications<ul style="list-style-type: none">- Briefly explore the deployment on Kubernetes with `kubectl` and show how applications are built up on Kubernetes (statefulset, pods, persistent volumes, etc)- Show relations in `juju status --relations`- Break the relation with `juju remove-relation`- Deploy grafana on MicroK8s and scale to three units, show the peer relation established in Juju status- Remove postgres from the model, redeploy in a different model and establish the relation using a cross-model relation	45 mins
Break	Break / Informal Q&A	10 min
Storage & Spaces	<ul style="list-style-type: none">- Explain the concepts around storage in Juju - storage pools, supported storage types- Explain how storage differs across LXD, MAAS, AWS/Azure/GCP and Kubernetes<ul style="list-style-type: none">- Tailor this section to the substrates the customer works with but provide an overview of other options- Introduce the `add-storage` and `remove-storage` commands- Explain how storage can be attached/detached from units at runtime<ul style="list-style-type: none">- Caveats for Kubernetes- Demonstrate how to add storage to a unit ,detach and reattach to a different unit- Introduce Juju Spaces	90 hr

	<ul style="list-style-type: none"> - Explain how spaces are supported in MAAS, EC2, Azure, Openstack - Demonstrate how to add a space - Deploy an application into the new space - Explain how constraints can be used to restrict models and applications to specific network spaces 	
Lab 9	<ul style="list-style-type: none"> - Create a new space, define a constraint on the model so that all new deployed applications are deployed into this space - Walkthrough this document: [Appendix] Installing Ceph - doc-Charmhub - Add 2 more storage volumes to Ceph OSD - Remove one of the volumes 	60 mins
Questions	Wrap up session Q&A	remainder