Juju Backup and Restore

Erlon R. Cruz | Sustainning Engineering

As of today, the Juju controller backup/restore process is partially supported. And restoring a controller to a different machine is not fully supported.

```
[ubuntu@sombrafam-bastion(kvm):~]$ juju help commands | grep backup
create-backup                 Create a backup.
download-backup               Download a backup archive file.
```

- `juju-restore` tool is off-tree and does not support restoring a controller to a different machine. You can only revert to the same machine (snapshot revert like).
- A community guide exists but is updated and requires a lot of editing and manual steps.
- We want something that can be as automated as possible and reliable.

1. Setting up the stage

Back up and shut off old controllers to prevent conflicts. Bootstrap a new controller with the same Juju version, noting all UUIDs and IPs.

```
# bootstraps a new controller
juju bootstrap <cloud>

# displays the required information
juju controllers --format json  | jq
juju models --format json | jq
```

Ensure essential variables are configured in the shell for consistent environment setup across all systems used.

🛠️ User interaction required

```
# On bastion
BACKUP_FILE_NAME=""

NEW_CONTROLLER_NAME=""
NEW_CONTROLLER_UUID=""
NEW_CONTROLLER_MODEL_UUID=""
NEW_CONTROLLER_INSTANCE_ID=""
NEW_CONTROLLER_IP=""

OLD_CONTROLLER_NAME=""
OLD_CONTROLLER_UUID=""
OLD_CONTROLLER_MODEL_UUID=""
```

2. Unpack the backup and copy it to the new controller:

```
# On bastion
juju model-config -m ${NEW_CONTROLLER_NAME}:controller logging-config='<root>=DEBUG;unit=DEBUG'
rm -rf juju-backup-old && tar -xvf $BACKUP_FILE_NAME && mv juju-backup juju-backup-old
pushd juju-backup-old && tar -xvf root.tar && popd
rsync -av  -e "ssh -i ~/.local/share/juju/ssh/juju_id_rsa" juju-backup-old ubuntu@${NEW_CONTROLLER_IP}:/tmp/
ssh ubuntu@${NEW_CONTROLLER_IP}
```

3. Preparing the new controller

- export all variables into the controller
- `yq` makes easier to edit config files

```
# Switch to root user to ensure all commands have the necessary permissions
sudo -s

# Export necessary environment variables defined previously (e.g., NEW_CONTROLLER_NAME)
# These variables may be required in subsequent steps
# NEW_CONTROLLER_NAME=
# ...
```

We also need to update the service files of the new controller to match those of the old controller. The service should be named after the old controllers' master machine number.

```
cd /etc/systemd/system
sudo mv jujud-machine-0.service jujud-machine-0.service.bak
sudo mv jujud-machine-0-exec-start.sh jujud-machine-0-exec-start.sh.bak
sudo cp jujud-machine-0.service.bak jujud-$machine_folder.service
sudo cp jujud-machine-0-exec-start.sh.bak jujud-$machine_folder-exec-start.sh
sudo sed -i "s/machine-0/$machine_folder/g" jujud-$machine_folder.service
sudo sed -i "s/machine-0/$machine_folder/g" jujud-$machine_folder-exec-start.sh
sudo sed -i "s/--machine-id 0/--machine-id $machine_id/g" jujud-$machine_folder-exec-start.sh

sudo systemctl daemon-reload
```

4. Testing database access

Start MongoDB with the right certificate, update agent.conf for SSL, and test connection with new credentials.

```
agent="machine-0"
yq eval ".cacert" "/var/lib/juju/agents/$machine_folder/agent.conf" > /var/snap/juju-db/common/ca.pem
sudo snap start juju-db

sudo juju-db.mongo --tlsCAFile /var/snap/juju-db/common/ca.pem --tls -u $agent -p $new_dbpass localhost:37017/admin
```

5. Restoring the database

Restore the mongo dump from the backup:

```
rm -rf /var/snap/juju-db/common/juju-backup-old/ && sudo cp -r /tmp/juju-backup-old/ /var/snap/juju-db/common/
sudo juju-db.mongorestore --ssl --sslCAFile /var/snap/juju-db/common/ca.pem \
  -u $agent -p $new_dbpass --authenticationDatabase admin --drop  -h localhost \
  --port 37017 --oplogReplay --batchSize 10 /var/snap/juju-db/common/juju-backup-old/dump/
```

6. Final steps before database manipulation

Before we leave the shell, let’s make our lives easier by printing the variables we’re going to use in Mongo. Make sure all of them are correct and set every time you need to re-connect to mongo.

```
echo "var newDbPass = '$new_dbpass'"
echo "var oldDbPass = '$old_dbpass'"
echo "var machineFolder = '$machine_folder'"
echo "var machineId = '$machine_id'"
echo "var newControllerModelUUID = '$NEW_CONTROLLER_MODEL_UUID'"
echo "var newControllerUUID = '$NEW_CONTROLLER_UUID'"
echo "var newControllerInstanceID = '$NEW_CONTROLLER_INSTANCE_ID'"
echo "var newControllerIP = '$NEW_CONTROLLER_IP'"
echo "var newControllerHostname = '`hostname`'"
echo "var oldControllerUUID = '$OLD_CONTROLLER_UUID'"
echo "var oldControllerModelUUID = '$OLD_CONTROLLER_MODEL_UUID'"
```

7. Database manipulations

All database manipulations going forward are based on the variable set before and doesn’t require manual interaction.

For example:

```
var newDbPass = ''
var oldDbPass = ''
var machineFolder = ''
var machineId = ''
var newModelControllerUUID = ''
var newControllerInstanceID = ''
var newControllerIP = ''
var newControllerHostname = ''
// ...
```

- 7.1. Create the agent’s user (machine-3, for example) copy roles
- 7.2. Reconnect to the new database using the new user (delete machine-0)
- 7.3. Fixing peergrouper information
- 7.4. Clean up the IP addresses for the restored agent.
- 7.5. Update the instance-id for the restored controller.

7.6. Cleanup the ip.addresses collection on the controller machine

```
var controller = db.machines.find({"jobs": 2})
controller.forEach(function(machine) {
    print("Updating machine addresses")
    db.machines.update({"_id": machine._id},
        {$set: {
            "addresses.0.value": newControllerIP,
            "machineaddresses.0.value": newControllerIP,
            "preferredpublicaddress.value": newControllerIP,
            "preferredprivateaddress.value": newControllerIP
    }})
})
// The addresses of the machine should point to the new controller IP
db.machines.find({"jobs": 2}).pretty()
```

7.7 Cleanup the IP addresses that are not being restored
7.8 Cleanup the machines that are not being restored

8. Start Juju Services

```
sudo systemctl start jujud-$machine_folder.service
```

9. Update the controller's configuration

Update the bastion's controller file (.local/share/juju/controllers.yaml) with the new controllers' api-endpoints address.

```
# On bastion

juju_share='.local/share/juju/controllers.yaml'
apiep0=$(sudo yq eval '.controllers.'$NEW_CONTROLLER_NAME'.api-endpoints[0]' $juju_share)
apiep1=$(sudo yq eval '.controllers.'$NEW_CONTROLLER_NAME'.api-endpoints[1]' $juju_share)

sudo yq eval '.controllers.'$OLD_CONTROLLER_NAME'.api-endpoints[0] "'$apiep0'"' -i $juju_share
sudo yq eval '.controllers.'$OLD_CONTROLLER_NAME'.api-endpoints[1] = "'$apiep1'"' -i $juju_share
```

10. Update the controller’s IP address in the agents

Update the controller’s IP address in the agents’ configuration files. You need to run this for each model.

** 🛠️ User interaction required **

```
# On bastion

model=""
if [ -z "$model" ]; then
  echo "Please set the model variable"
  exit 1
fi

for m in `juju status -m $model --format=json | jq -r '.machines | keys | join("\n")'`; do
  echo "Fixing machine address  for model ${model}" machine-$m
  juju ssh -m $model $m 'cd /var/lib/juju/agents; for a in `ls -d *`; do echo "replacing $a/agent.conf" ; sudo sed -i
  # juju ssh -m $model $m 'cd /var/lib/juju/agents; for a in `ls -d *`; do echo "replacing $a/agent.conf" ; sudo sed
  juju ssh -m $model $m "sudo systemctl restart jujud-machine-${m}"
done
```

11. Enable HA

As of now, all your models should be reachable and the agents in the units from each model in the same status as they were before the backup restore. Make sure that that is the case. If everything is working as expected, you can enable HA.

```
juju enable-ha
```

Needs testing scenarios

1. Test the new controller by deploying a new model and a new application.
2. Test the migration of multiple models and applications.
3. Adding new applications to the model after the restoring process.
4. Deleting the models and applications after the restoring process.
5. Test on Focal
6. Test on 3.5.x
7. LXD deployments

Additional Resources

- Google Docs Version for Contributing
- Rendered Markdown With the Full Guide

Special Thanks

- Alan Baghumian
- Nicolas Bock
- John Meinel & Juju Team