Ansible SpeckleServer Role

ansible-role-speckleserver

This role installs SpeckleServer, a design and AEC data communication and collaboration platform.

Requirements

This is primarily designed to run on a Linux host, with a Redis cache and MongoDB NoSQL database. These can be provided externally (for instance in AWS or other cloud providers) or via other Ansible roles.

Supported Operating Systems

- Red Hat Enterprise 7 (including CentOS 7)
- Ubuntu LTS 16.04 ("xenial") and 18.04 ("bionic")
- Amazon Linux 2
- Debian 8 ("jessie") and 9 ("stretch")

Role Variables

`speckleserver_version`: Version of SpeckleServer to download and install. Can be a numbered version in major.minor.micro format or "latest" (default: `latest`)

`speckleadmin_version`: Version of the SpeckleAdmin plugin to download and install. Can be a numbered version in major.minor.micro format

`speckleserver_server_name`: A name for this SpeckleServer instance.

`speckleserver_url`: The canonical web URL for this SpeckleServer instance (default: `http://localhost/`)

`speckleserver_public_streams`: Allow streams to be publicly published (default: `true`)

`speckleserver_port`: Port for SpeckleServer to listen on (default `3000`)

`speckleserver_listen_ip`: IP address to bind to (defaults to `127.0.0.1`)

`speckleserver_session_secret`: Session cookie secret: (default "`changemeplease`")

`speckleserver_mongodb_uri`: MongoDB connection string: (default `mongodb://localhost:27017/`)

`speckleserver_redis_host`: Redis hostname to connect to (default `localhost`)

`speckleserver_redis_port`: Redis port (default `6379`)

`speckleserver_pretty_json`: Pretty print API response output. Note that this makes the responses 10% larger. Boolean value, default is `false`.

`speckleserver_expose_emails`: Makes all user email addresses publicly visible. Not recommended for production. Boolean value, defaults to `false`.

Dependencies

There are no hard requirements as such aside a webserver to proxy the application traffic.

I would strongly suggest running this alongside a webservice role such as Jeff Geerling's Apache module (`geerlingguy.apache`) or NGINX Inc's nginx module (`nginxinc.nginx`)

Let's Encrypt's Certbot is also useful if you need SSL/TLS security on a public-facing instance (again, consider `geerlingguy.certbot` from Jeff Geerling)

If you need a MongoDB instance alongside this, I'd recommend `undergreen.mongodb` (covers RHEL/CentOS as well as Debian/Ubuntu). This is particularly useful in AWS regions where DocumentDB is not available.

These are all easily installable via the `ansible-galaxy` command (part of the base Ansible package) - eg. `ansible-galaxy install geerlingguy.nginx`

How to run

Ensure you have Ansible set up on your machine appropriately and the role is unpacked in a directory listed in roles_path

A playbook for setting up with nginx is provided: (`speckleserver-with-nginx-playbook.yml`).

Copy this to a new file eg. site-speckle.yml; tune the vars as required.

WARNING: Hosts / Virtual hosts in this playbook are only given as placeholders and will not work out-of-the-box. Customise them before running the playbook!

Note: You can also cut and drop the vars: stanza into group_vars/ or host_vars as you require.

Test the playbook

```
ansible-playbook -C site-speckle.yml
```

If it retrurns OK, run the play

```
ansible-playbook site-speckle.yml
```

Testing / Development

A Molecule test plan is provided.

You will need a local Docker setup and the Molecule test suite installed via pip (`pip install --user molecule`).

You will also need a functional Ansible setup of course :D

`molecule check` performs a dry test run

`molecule test` performs the full test suite

If you wish to test cross-distro, pass a `MOLECULE_DISTRO` environment variable in your molecule command eg. `MOLECULE_DISTRO=ubuntu1804 molecule test`

Supported distros for such testing, courtesy of Jeff Geerling's molecule-testing docker images:

- CentOS 7 (`centos7` - the default)
- Ubuntu Bionic LTS 18.04 / Xenial LTS 16.04 (`ubuntu1804` / `ubuntu1604`)
- Debian 8 and 9 (`debian8` / `debian9`)

TODO

- More idempotency / upgrade checks and tests.
- Resilience in the face of URL fetch failures

Example Playbook

```
- hosts: servers
  become: yes
  become_method: sudo
  become_user: root
  roles:
     - { role: ansible-role-speckleserver  }
  vars:
    speckleserver_server_name: 'SpeckleServer Development Environment'
    speckleserver_url: 'http://speckle.example.com'
    speckleserver_mongodb_uri: 'mongodb://mongo.example.com:27017/speckle'
    speckleserver_redis_host: 'redis.example.com'
    speckleserver_version: "1.5.2"
    speckleadmin_version: "0.2.11"
```

License

BSD

Author Information

Michael Fleming michael.fleming@arup.com