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_admin`: Install the SpeckleServer Admin interface from SpeckleWorks SpeckleAdmin git. Boolean value, default is false.

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

`speckleserver_server_name`: A name for this SpeckleServer instance.

`speckleserver_url`: The canonical web URL for this SpeckleServer instance (default: `http://localhost`) without the trailing slash.

`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_max_procs`: [**OPTIONAL**] number of SpeckleServer processes to spawn (no default, software will spawn a process per CPU core if left unset)

`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`.

`speckleserver_request_max` : Maximum request size the server will accept, to protect against floods / DoS. defaults to `10mb`.

`speckleserver_first_user_admin` : The first created user in a new Speclke-Server 1.6.x or later instance is considered the admin user. Boolean value, default is `true`.

`speckleserver_send_email`: The application (1.6.x and later) can send email notifications to users. Boolean value, defaults to `true`

Email related options

- `smtp_host`: mailserver hostname. String value, default is `localhost`
- `smtp_port`: mailserver port to use. Integer value, default is `25`, Recommended values are `25` (standard smtp) or `587` (submission) if available
- `smtp_auth_username`: SMTP Authentication username (if supported). String value, Defaults to an empty string (no AUTH)
- `smtp_auth_password`: SMTP Authentication password (if supported). String value, Defaults to an empty string (no AUTH)
- `smtp_email_sender`: SMTP envelope sender address. String value, default is `speckleserver` (your MTA should append the local domain/FQDN if not overriden)

`speckleserver_public_registration`: Allows user registration via the application directly or administrative frontend. Boolean value, default is true

`speckleserver_local_auth`: Use local, internal user authentication (includes password resets when email enabled). Boolean value, default is true

`speckleserver_wl_redirect_urls`: A comma-separated list of URIs we will accept for redirects - note that localhost is implied. String value, empty set for defaults

`speckleserver_allow_insecure_redirects`: Allow insecure, unencrypted redirects. Not recommended unless you know what you're doing. Boolean value, sensibly defaults to false `speckleserver_api_version`: Declare the supported SpeckleServer API version. Should be `1.x.x` for 1.9.x, and `2.x.x` for 2.x and above. String value, defaults to `1.x.x`

External User Authentication

As of 1.7, Speckle can also authenticate users with the following three services and variable sets : Auth0 , Azure Active Directory and Github user authentication

Each has to be toggled on (boolean true/false) and a YAML hash of settings provided, per the below reference. Precise details are left as an exercise for the reader.

As a general rule, you'll need a client identifier, the associated secret and any relevant metadata (domain names, callback URLs, IdP metadata stanzas) per your chosen PaaS provider.

Auth0:

```yaml
# Auth0 needs a client ID / Secret and a domain the account is attached to
speckleserver_auth0_auth: true
speckleserver_auth0:
  clientid: "my_auth0_client_id"
  domain: "my_dns_domain.org"
  clientsecret: "my_auth0_client_secret"
```

Microsoft Azure AD:

```yaml
# Azure needs the client id/secret, the AzureAD organization name and the Tenant ID for some
speckleserver_azuread_auth: true
speckleserver_azuread:
  orgname: "my_organization_name"
  clientid: "my_client_id"
  metadata: "my_azuread_tenant_id"
  clientsecret: "my_azuread_client_secret"
```

GitHub:

```yaml
# Github user auth requires the client ID/Secret and a callback URI
speckleserver_github_auth: false
speckleserver_github:
  clientid: "my_github_org_client_id"
  clientsecret: "my_github_org_client_secret"
  callback: "my_github_callback_url"
```

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

Copy the `site.yml` file to a new file eg. `site-speckle.yml`; tune the hosts and vars stanzas as required.

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

- Better vhost examples (nginx vhost behind an SSL-terminating loadbalancer for example)

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"
```

Nginx webserver integration

You will need the `nginxinc.nginx` role from Galaxy (this should be installed per requirements.yml anyway :D)

An example playbook: speckle-plus-nginx.yml

You can of course move the vars into your group_vars entry alongside the speckle_* settings if you prefer.

License

Author Information

Michael Fleming michael.fleming@arup.com