SLATE Application Development and Review Procedures

Version	Comment	Effective date
1.0	Initial version	September 25, 2020
2.0	Updated review procedures	March 23, 2021

Preface

Applications intended for deployment and operation using the SLATE platform differ from standard Kubernetes applications in a number of ways. Given SLATE's role in providing an edge services platform, the cybersecurity requirements must align and satisfy those of Edge Cluster host institutions. In particular, applications (often containerized services) must adhere to the following:

- They have a well-understood image provenance. Images are typically:
    - Provided for use in SLATE by a "Trusted Organization" (see the Trusted Image Sources document)
    - Developed or maintained directly by the SLATE team, and with sources stored in source control repositories overseen by the SLATE Team
    - Have all dependent image layers also follow the previous two rules
- Must not allow Application Administrators to substitute container image(s) used by the application.
- Must be accompanied by a README document as described below to provide descriptive, implementation related, and review information, especially for Edge Administrators and Application Administrators.

Application Development

Application Developers should submit new applications and modifications as pull requests to the SLATE stable catalog (https://github.com/slateci/slate-catalog).

Packaged applications must have a Helm chart which defines how the application is installed in Kubernetes, and may also contain the sources for container images used by the application.

This document is a policy of the SLATE (Services Layer at the Edge) project, supported by the National Science Foundation Office of Advanced Cyberinfrastructure: "CIF21 DIBBs: EI: SLATE and the Mobility of Capability", award number OAC-1724821.

Unless an image is drawn from a source on SLATE's allow-list of trusted external image maintainers/sources, its sources must be included in the catalog with the chart which uses it.

The chart must have a README document (for which the template is provided in the Appendix) which provides guidance on what the application is for and how to install and use it. The README is divided into two main sections, one to be completed by the Application Developer and the other by the Application Reviewer. The Developer completed section may contain a reference or link to more complete documentation for the application that is maintained elsewhere. When possible, Application Developers should provide guidance on how to test the functioning of the application with widely, or at least publicly, available tools.

The default configuration of an application should, as much as possible, be chosen to be reasonably safe to operate (e.g. proxies should not default to being open to the entire internet, etc.). Any features of the application which have significant security implications must be clearly mentioned in the README with explanations of the concerns an operator should be aware of.

Application developers are encouraged to test their applications by installing them directly to a Kubernetes cluster (such as minikube) with Helm, or in the miniSLATE test environment to ensure proper functionality before submitting them for review, as this makes the review more efficient for all parties.

Application Review

Each item to be put into the SLATE stable catalog must first undergo the review process defined in this document. The item might consist of one or possibly more containers, all of which are referenced in a single Helm chart. We'll refer to this item as an application throughout this document.

Every application must be reviewed by at least one person not responsible for its development or packaging. The README document in the Appendix contains a list of questions for the review. At the end of each review, the Application Reviewer will update the Reviewer completed section of the README document with the review findings, and make it available along with the container in the stable catalog. This approach allows stakeholders to see application specific information and the associated review for the referenced version. Any version update of the application will require an updated README document.

Application Maintenance

Applications previously reviewed by an Application Reviewer and curated into the SLATE stable catalog can receive regular updates for maintenance purposes, feature updates, security updates, etc. All of the principles and obligations of the Application Development section above apply.

Any change to an application must include an update of the chart's version number. This is necessary to ensure that versions are distinguishable, and that Helm will actually use the new version, rather than a stale one from its local cache.

Each change to an application must be successfully reviewed before being added to the SLATE stable catalog. When there are only a few minor changes, the Reviewer and Developer may decide to focus the process defined here only on these changes.

Appendix - README Template

README template introduction

The developer of the application must create a README for their application which resides in the corresponding git repository. The README comprises two main sections. The first is to be completed by the Application Developer, while the second is to be completed by the Application Reviewer:

- Developer section:
    - Description of what the application is and does
    - Description of how to install the application
    - Description of network requirements
    - Description of storage requirements
    - Description of application service stateful requirements
    - Description of service privilege requirements, i.e. secrets
    - Description of monitoring and alerting requirements
    - Description of service testing and basic functionality validation

- Reviewer section:
    - Set of questions for a reviewer to validate with the developer before approval into the SLATE stable catalogue

The Reviewer's section must stay with the README as part of the document so that users and groups deploying the application are fully informed by the application review process. Please feel free to omit responses in sections that do not apply. Each released version of the application should have an updated README document associated with the application.

README Section to be Completed by Developers

Application Name and Version

Please provide the specific name(s) and version(s) of the application under review. These must correspond to analogous information in the Helm chart.

Description

Application Developers should write up a description of the application and note at least the following information.

- What does this application do?
- What is the context for use of this application?
- Is it specific to a particular Virtual Organization or community?
- Are there any specific use cases that this application addresses or that Edge Administrators, Application Administrators, or users might want to know?

Installation

- What are the steps needed to install this application?
    - Images?
    - YouTube video references?
    - Other references?
    - See below sections for things to document
- Are there versioning considerations of which to be aware? Are multiple versions of deployments a possibility and recommended?
- Clearly define application-specific secret structure so that Application Administrators can properly implement it.
- Does the application require special resources such as special hardware, software credentials or other unique resources for validating functionality?

System requirements

Please describe the recommended system resources required by the pod for the instance of the set of services that will deploy.

Description	**Answers and Special Notes**
What are the CPU requirements, i.e. how many cores and at what speeds?	
What are the memory requirements?	
Describe any special resources required by the application.	

Network requirements

Please describe the application's network requirements.

Description	**Answers and Special Notes**

What are the TCP/UDP ports the application needs to expose to the Internet? (i.e. http, https, specific port ranges)	
Will this application be an IPv4 service?	
Will this application be an IPv6 service?	
Does the service require its own IP?	
Does the service utilize/require a load balancer, i.e. metallb	

Storage Requirements

Please describe the application's storage requirements.

Description	**Answers and Special Notes**
From where does the service expect storage? I.e. local storage, Amazon S3, Ceph with RADOS gateway, POSIX NFS, CIFS, CephFS, Block	
Does the service need to mount storage across multiple deployments locally or geographically? I.e. across 2 local SLATE deployments or across multiple SLATE deployments across the US	
How much storage does your service need at each deployment site?	
Does the storage require specific performance characteristics? I.e. does it require SSD, 15k SATA disk or something else?	

Statefulness

Please describe the statefulness of the application and whether it relies on additional services external to it. Please describe any dependencies.

Description	Answers and Special Notes
Is the service stateful or stateless? Example: does the service require the ability to resume a session, or, is each instance of the service completely independent if it fails or is suddenly shutdown?	
Does the service require other services in the same Pod, or, is it completely stand-alone? Example: webserver with transactional database	

Privilege requirements

Please describe the privileges, role requirements, and security attributes this pod of services requires at each site.

Description	Answers and Special Notes
Does the service require specific certificates?	
What are the requirements of the certificates, i.e. do they require dissemination from a central source on a regular basis?	
Does the service require any specific key/value attributes regarding security requirements?	
Does the application require any specific privilege mode? Example: Role RBAC. *NOTE: SLATE does not allow ClusterRole RBAC for any application.*	
Does the application require any container-root access to the host filesystem?	
Does the application require user-level access within the container? Avoid if possible since some site security policies may prohibit.	

Does the application require “host networking”? If so, why? Avoid if possible for greatest flexibility in deployment across nodes.	
Does the application use container storage simply as a path to the host’s storage? If so, why? Avoid if possible.	
Does the application have any configuration that might remove or modify the basic container isolation?	
Does a network policy template exist in the Helm chart? Is it default-deny? Will the App Administrator or Edge Administrator need to modify the container’s policy? Does the template contain any needed parameters that Application Administrators will specify?	
Does all user file access occur in a container’s non-root file system?	
Does each container have its own, exclusive root file system?	
Does the containerized app have a read-only root file system?	

Labels and Annotations

Description	**Answers and Special Notes**
Does the Helm chart include the SLATE recommended labels and produce corresponding Kubernetes objects? (ref …)	

Monitoring and Logging

Please describe the monitoring and logging requirements.

Description	Answers and Special Notes
Does the pod of services require central monitoring, or is its output directed to an external dedicated monitoring infrastructure?	
Does the pod of services require special ports or considerations for monitoring?	
Does the pod of services require additional sidecars for logging manipulation or local logging facilities?	

Multiple Versions

Description	Answers and Special Notes
Does the application require the ability to support multiple versions simultaneously?	

Testing

Please describe the functionality testing performed and any particular support for operational testing, eg, trying it out in a sandbox environment prior to whitelisting it.

Description	**Answers and Special Notes**
Does the application include any specific testing framework, set of unit tests, or testing scripts for functionality validation?	
Does the application include any operational tests to validate Kubernetes deployment status, open network ports, resource utilization, etc.?	

README Section to be Completed by Reviewer

The Reviewer should validate that the above Developer completed README is complete (as applicable) and comprehensible. If the Reviewer has questions regarding the README, so will users. Please note these questions to the Application Developer in order to have clarity. In addition to validating information in the Developer's README section, the Reviewer should ask the following questions. If the Application Developer has gone over these questions already and has pointers, then they should include pointers to these in the Reviewer section for the reviewer to consider.

Review Questions	**Answers and any notes**
Does output of helm lint show any errors? What are the changes to fix flagged errors?	
Does the origin of each container image used in the application comply with the image provenance rules?	
Does the helm chart produce Kubernetes objects with appropriate names that distinguish release, instance, etc. in order to prevent collisions between multiple simultaneous instances?	
Does the application require special resources such as special hardware, software credentials or other unique resources for validating functionality? ● If yes, does the Developer section of this Readme document adequately describe these requirements? ● If yes, is there a process by which a reviewer can work with the application developer to validate the installation in an appropriate environment?	
Has the application developer worked with the reviewer to scan the application for vulnerabilities? ● If yes, has the application developer been able to mitigate 'critical' or 'high' severity issues? ● Are the results of the scans able to be public?	
Are the application developers and administrators available for questions, consultation, and recommendations during the review process?	
Will the review of the application be able to be public? If no, please explain the reasons, i.e. intellectual property, journal publishing, sensitive data, etc.	

Does the container handle sensitive data? NOTE: SLATE does not support the separation of public and private data, nor does it have the provision to handle sensitive data. A group may roll their own implementation of SLATE and its catalogue based on the current reference system, if desired.	
Please write any additional Reviewer questions here.	

Vulnerability Assessment

Conduct a scan of the application with a container scanning tool such as Clair (https://github.com/quay/clair). Discuss prospect for remediation with the Developer of any vulnerabilities identified as critical or high. Provide below a summary of the scan results followed by the fully detailed report. If the Developer remediates any vulnerabilities prior to resubmitting their application for the SLATE stable catalog, re-run the scan to ensure that the scan results included in this README accompanying the application in the SLATE stable catalog was run on that version of the application.