

From Source Code to Kubernetes Manifests: Guiding LLMs with Structured Metadata

Matteo Franzil^{1,2}
matteo.franzil@unitn.it

Ulises Emiliano Sosa¹
ulises.e.sosa@proton.me

Domenico Siracusa¹
domenico.siracusa@unitn.it

¹ Department of Information Engineering and Computer Science, University of Trento, Italy
² Center for Cybersecurity, FBK - Fondazione Bruno Kessler, Trento - Italy



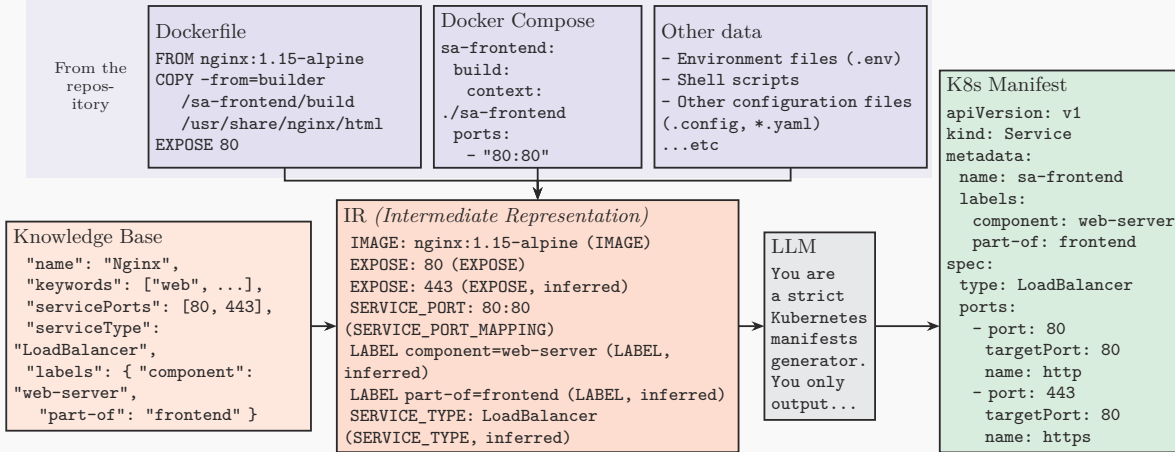
Introduction



Deploying applications for **Kubernetes** requires writing **manifests**: complex YAML files that define resources and configurations. This requires deep expertise in both the application architecture and Kubernetes concepts. **Large Language Models (LLMs)** offer promise for automation, but struggle without proper context.

Our **pipeline** extracts metadata from repositories, structures it into an **Intermediate Representation (IR)**, and uses it to guide LLMs in generating accurate, deployable manifests.

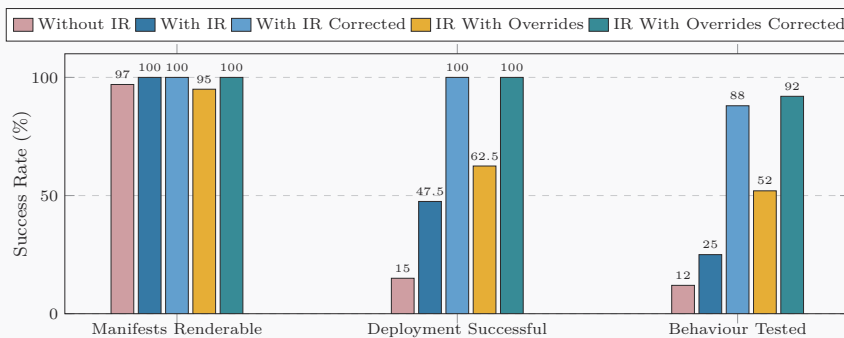
Architecture



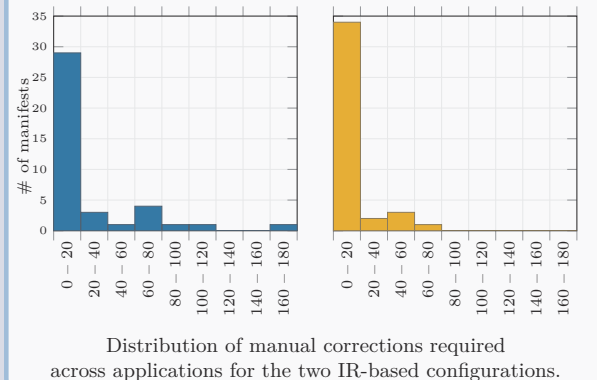
IR mappings

Dockerfiles:
CMD/ENTRYPOINT
Pod command
EXPOSE
containerPort
ENV
ConfigMap/Secret
VOLUME
PersistentVolume
WORKDIR
workingDir
USER
[...]runAsUser
HEALTHCHECK
Readiness/Liveness probes
Docker Compose:
services
Deployment/Service
depends_on
Init containers/Probes
networks
Service connectivity
ports
NodePort/LoadBalancer

Deployability



Human Effort



Typical Errors

<pre>apiVersion: v1 kind: Service metadata: name: result spec: type: LoadBalancer selector: app: result ports: - port: 9229 targetPort: 9229 name: debug selector: app: result</pre>	<pre>apiVersion: v1 kind: Service metadata: name: productpage spec: type: ClusterIP selector: app: productpage ports: - port: 9080 targetPort: 9080 protocol: TCP [...]</pre>
--	---

Example of two common errors in LLM-generated manifests.

Semantic Alignment

Stage	H	L
Without IR	12	95
With IR	25	92
IR Corrected	88	92
IR + Overr.	52	98
IR + O. Corr.	92	95

H: Human-validated functional correctness
L: LLM-judged alignment w.r.t. requirements
(all values in %)

Conclusions

Structured metadata significantly improves LLM performance in generating Kubernetes manifests, demonstrating the value of targeted context over raw code dumps or generic prompts. However, **human oversight remains essential** for complex deployments.

Future directions:

- Expand evaluation to other LLMs
- Implement iterative feedback loops for self-correction
- Extend security evaluation of images and manifests